

# **Modulverzeichnis**

der Studiengänge

Angewandte Informatik (B.Sc.) – BIN  
Angewandte Informatik (M.Sc.) – MIN

an der Hochschule Hannover

## Inhaltsverzeichnis

<b>Module und Dozentinnen/ Dozenten des Bachelorstudiengangs</b> .....	<b>3</b>
<b>Module und Dozentinnen/ Dozenten des Masterstudiengangs</b> .....	<b>4</b>
<b>Bachelorstudiengang Angewandte Informatik – Modulbeschreibungen</b> .....	<b>5</b>
Grundlagen der Informatik.....	6
Rechnerstrukturen .....	7
Theoretische Informatik.....	8
Datenstrukturen & Algorithmen .....	9
Datenbanken .....	10
Betriebssysteme / Unix.....	11
Programmieren I (Einführung in die Programmierung: Java).....	12
Programmieren II (Objektorientierte Programmierung) .....	13
Programmieren III.....	14
Programmierprojekt.....	15
Mathematik II.....	17
Grundlagen der Betriebswirtschaftslehre .....	19
Ergänzende Fächer I.....	20
Software Engineering I .....	21
Software Engineering II .....	22
Software Engineering III .....	23
Informationssysteme I .....	24
Informationssysteme II .....	25
Betriebssysteme und Netze I .....	26
Betriebssysteme und Netze II .....	27
Computergraphik I (Generative Computergraphik) .....	28
Computergrafik II (Bildverarbeitung) .....	29
Seminar .....	30
Wahlpflichtfach Informatik .....	31
Ergänzende Fächer II.....	32
Praxisprojekt I.....	33
Praxisprojekt II.....	34
Bachelorarbeit mit Kolloquium.....	35
<b>Masterstudiengang Angewandte Informatik – Modulbeschreibungen</b> .....	<b>36</b>

## Module und Dozentinnen/ Dozenten des Bachelorstudiengangs

Bachelorstudiengang Angewandte Informatik				
Lfd. Nr.	Modul	Abkürzung	Modulverantwortlichkeit	Dozentinnen/ Dozenten
1	Grundlagen der Informatik	GDI	Ahlers	Ahlers, Laumann
2	Rechnerstrukturen	RS	Ahlers	Ahlers, Laumann
3	Theoretische Informatik	TI	Kleiner	Kleiner, Sprengel
4	Datenstrukturen und Algorithmen	DA	Kleiner	Ginkel, Hovestadt, Kleiner, Peine, Wohlfeil
5	Datenbanken	DB	Dennert-Möller	Dennert-Möller, Kleiner, Koschel
6	Betriebssysteme/Unix	BS	von Helden	von Helden, Hovestadt, Müller, Peine
7	Programmieren I (Einführung in die Progr.)	PR1	Bruns	Bruns, Dunkel, Holitschke, Kleiner, Wohlfeil, Peine
8	Programmieren II (Objektorientierte Progr.)	PR2	Dunkel	Bruns, Dunkel, Holitschke, Peine
9	Programmieren III (C/C++)	PR3	Klingenberg	Hovestadt, Peine
10	Programmierprojekt	PP	Klingenberg	Holitschke, alle Lehrenden
11	Mathematik I	MAT1	Dennert-Möller	Ahlers, Dennert-Möller, Sprengel
12	Mathematik II	MAT2	Sprengel	Ahlers, Dennert-Möller, Sprengel
13	Mathematik III (Numerik und Statistik)	MAT3	Dennert-Möller	Ahlers, Dennert-Möller, Sprengel
14	Grundlagen der Betriebswirtschaftslehre	BWL	von Helden	LB
15	Ergänzende Fächer I	EF1	Klingenberg	LB
16	Software Engineering I	SE1	Bruns	Bruns, Dunkel, Koschel, Peine
17	Software Engineering II	SE2	Bruns	Bruns, Dunkel, Koschel, Peine
18	Software Engineering III	SE3	Dunkel	Bruns, Dunkel, Koschel, Peine
19	Informationssysteme I	IS1	Kleiner	Dennert-Möller, Kleiner, Koschel
20	Informationssysteme II	IS2	Koschel	Bruns, Dunkel, Kleiner, Koschel
21	Betriebssysteme und Netze I	BSN1	von Helden	von Helden, Hovestadt, Wohlfeil
22	Betriebssysteme und Netze II	BSN2	Wohlfeil	von Helden, Hovestadt, Peine, Wohlfeil
23	Computergraphik I (Generative CG)	CG1	Sprengel	Ahlers, Sprengel
24	Computergraphik II (Bildverarbeitung)	CG2	Dennert-Möller	Ahlers, Dennert-Möller, Sprengel
25	Seminar	BSEM	Wohlfeil	Alle Lehrenden
26	Wahlpflichtfach Informatik	WPI	Koschel	Koschel, Wohlfeil
27	Ergänzende Fächer II	EF2	Klingenberg	LB
28	Praxisprojekt I	BPR1	Bruns	Alle Lehrenden
29	Praxisprojekt II	BPR2	Bruns	Alle Lehrenden
30	Bachelorarbeit mit Kolloquium	BA	Kleiner	Alle Lehrenden

## Module und Dozentinnen/ Dozenten des Masterstudiengangs

Masterstudiengang Angewandte Informatik				
Lfd. Nr.	Modul	Abkürzung	Modulverantwortlichkeit	Dozentinnen/ Dozenten
<b>Gemeinsamer Block</b>				
1	Softwarearchitekturen	MIN-SWA	Dunkel	Bruns, Dunkel, Koschel
2	Projekt- und Qualitätsmanagement	MIN-PMQM	Koschel	Bruns, Hovestadt, Koschel
3	Visualisierung und HCI	MIN-VISH	Ahlers	Ahlers, Ginkel
4	Algorithmen und Komplexität	MIN-AK	Ginkel	Ginkel, Sprengel, Heine, Kleiner, Garmann
5	Seminar	MIN-MSEM	Wohlfeil	Alle Lehrenden
6	Masterprojekt	MIN-MSPR	Dunkel	Alle Lehrenden
7	Masterarbeit	MIN-MA	Wohlfeil	Alle Lehrenden
<b>Schwerpunkt Computergrafik</b>				
G8	Geometrische Modellierung	MIN-GM	Sprengel	Sprengel, Ginkel
G9	Computergrafik und Interaktion	MIN-CGI	Ahlers	Ahlers, Sprengel, Ginkel
G10	Visual Computing	MIN-VC	Sprengel	Ahlers, Dennert-Möller, Sprengel, Ginkel
G11	Computational Geometry	MIN-COG	Ginkel	Ginkel, Ahlers, Sprengel
G12	Computer Vision	MIN-CV	Dennert-Möller	Dennert-Möller, Sprengel
<b>Schwerpunkt IT-Sicherheit</b>				
T8	IT-Sicherheit I	MIN-ITS1	Wohlfeil	von Helden, Wohlfeil, Peine, Hovestadt
T9	IT-Sicherheit II	MIN-ITS2	Wohlfeil	von Helden, Wohlfeil, Peine, Hovestadt
T10	Spezialthema IT-Sicherheit	MIN-SSI	von Helden	von Helden, Wohlfeil, Peine
T11	Secure Software Engineering	MIN-SSE	Peine	Peine, Wohlfeil Gleiches Modul wie S12, kann entweder in IT-Sicherheit oder in Softwaretechnik belegt werden
<b>Schwerpunkt Informationssysteme</b>				
I8	Datenbankparadigmen	MIN-DBP	Heine	Heine, Kleiner, Koschel, Dennert-Möller
I9	Verteilte Informationssysteme	MIN-VIS	Koschel	Koschel, Heine, Hovestadt
I10	Datenanalyse	MIN-DA	Heine	Dennert-Möller, Heine, Kleiner, Koschel
I11	Geo- und Multimedia Informationssysteme	MIN-GMI	Kleiner	Dennert-Möller, Heine, Kleiner
<b>Schwerpunkt Softwaretechnik</b>				
S8	Programmierparadigmen	MIN-PPD	Peine	Peine, Garmann, Kleiner, Wohlfeil
S9	Intelligente Systeme	MIN-INS	Bruns	Bruns, Dunkel, Wohlfeil
S10	Fortgeschrittene Aspekte verteilter Systeme	MIN-FAVS	Koschel	Koschel, Heine, Hovestadt
S11	Software-Test und Requirements	MIN-STR	Garmann	Garmann, Koschel, Peine
S12	Secure Software Engineering	MIN-SSE	Peine	Peine, Wohlfeil Gleiches Modul wie T11, kann entweder in IT-Sicherheit oder in Softwaretechnik belegt werden

**Bachelorstudiengang Angewandte Informatik – Modulbeschreibungen**

<b>Modulbezeichnung:</b>	<b>Grundlagen der Informatik</b>
<b>Kürzel:</b>	GDI
<b>Studiensemester:</b>	1. Semester (WS), Turnus: jährlich
<b>Modulverantwortliche(r):</b>	Volker Ahlers
<b>Dozent(in):</b>	Volker Ahlers, Bernd Laumann
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor, Pflichtfach, 1. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung = 68 h Übung = 34 h Eigenstudium = 108 h
<b>Kreditpunkte:</b>	7 CP (= 210 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	keine
<b>Angestrebte Lernergebnisse:</b>	<p>Algorithmische und mathematische Kompetenzen: Verständnis der mathematischen Grundlagen der Zahlendarstellung, der Kodierungstheorie und der Schaltungslogik; Berücksichtigung der Zahlendarstellung bei der Programmierung; Fähigkeit, einfache Probleme zu analysieren und algorithmisch zu lösen.</p> <p>Analyse-, Design- und Realisierungskompetenzen: Fähigkeit, WWW-Inhalte zu erstellen; Fähigkeit, logische Schaltungen zu entwerfen; Fähigkeit, algorithmische Lösungen zu programmieren.</p> <p>Technologische Kompetenzen: Kenntnis des technischen Aufbaus eines Computers; Erfahrung im Umgang mit Computer-Hardware; Kenntnis von Internet-Technologien; Verständnis der Funktionsweise von Robotern.</p>
<b>Inhalt:</b>	<p>Überblick über die Informatik: Teilgebiete, geschichtliche Entwicklung.</p> <p>Internet und WWW: Netzwerk-Grundlagen, HTTP, HTML, CSS.</p> <p>Zahlendarstellung und Kodierungstheorie: Dual-, Oktal-, Hexadezimalsystem, Gleitkommadarstellung, Zeichenkodierung, fehlertolerante Codes, Datenkompression.</p> <p>Technische Informatik: Digitaltechnik, Schaltungslogik, Hardware-Komponenten, Von-Neumann-Architektur, Peripheriegeräte.</p> <p>Einführung in Algorithmen: Analysieren und algorithmisches Lösen einfacher Probleme, Aufwandsabschätzung, Programmierübungen.</p> <p>Robotik: Sensoren, Aktoren, Konstruktions- und Programmierübungen.</p>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung von Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung</p>
<b>Literatur:</b>	<p>Herold, H., B. Lurz, J. Wohlrab: Grundlagen der Informatik. Pearson, neueste Auflage.</p> <p>Hoffmann, D.W.: Grundlagen der Technischen Informatik. Hanser, neueste Auflage.</p> <p>Becker, B., P. Molitor: Technische Informatik. Oldenbourg, neueste Auflage.</p>

<b>Modulbezeichnung:</b>	<b>Rechnerstrukturen</b>
<b>Kürzel:</b>	RS
<b>Studiensemester:</b>	2. Semester (SS), Turnus: jährlich
<b>Modulverantwortliche(r):</b>	Volker Ahlers
<b>Dozent(in):</b>	Volker Ahlers, Bernd Laumann
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor, Pflichtfach, 2. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 52 h
<b>Kreditpunkte:</b>	4 CP (= 120 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	Grundlagen der Informatik, Programmieren I
<b>Angestrebte Lernergebnisse:</b>	<p>Algorithmische Kompetenzen: Berücksichtigung der Prozessorarchitektur und der Speicherhierarchie bei der Programmierung.</p> <p>Design- und Realisierungskompetenzen: Fähigkeit, Assemblerprogramme zu verstehen und neu zu entwickeln.</p> <p>Technologische Kompetenzen: Verständnis der Funktionsweise von Mikroprozessoren und Computern; Kenntnis typischer Rechnerarchitekturen, inkl. eingebetteter Systeme; Kenntnis von Methoden zur Leistungssteigerung, inkl. Parallelisierung.</p> <p>Methodenkompetenzen: Fähigkeit, die Leistung von Prozessoren und Computern zu bewerten und zu vergleichen.</p>
<b>Inhalt:</b>	<p>Mikroprozessoren: ALU, FPU, Register, Steuereinheit, Datenpfad.</p> <p>Rechnerarchitektur: Von-Neumann-Architektur, CISC- und RISC-Architektur, Mikroprogrammierung, Pipelining, Interrupts, Bussysteme, I/O-Schnittstellen, Parallelisierung, Mehrkernprozessoren, eingebettete Systeme, Leistungsbewertung und Benchmarks.</p> <p>Speicherverwaltung: Speicherhierarchie, virtueller Speicher, Cache.</p> <p>Assembler-Programmierung: Maschinensprache, Befehlssatz, Mnemonics, Speicheradressierung, Stack, Unterprogramme, Betriebssystem- und I/O-Schnittstellen, Programmierübungen.</p>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung von Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung</p>
<b>Literatur:</b>	<p>Patterson, D.A., J.L. Hennessy: Rechnerorganisation und -entwurf. Spektrum, neueste Auflage.</p> <p>Tanenbaum, A.S., J. Goodman: Computerarchitektur. Pearson, neueste Auflage.</p> <p>Duntemann, J.: Assembly Language Step by Step. Wiley, neueste Auflage.</p>

<b>Modulbezeichnung:</b>	<b>Theoretische Informatik</b>
<b>Kürzel:</b>	TI
<b>Studiensemester:</b>	1. Semester (WS), Turnus: jährlich
<b>Modulverantwortliche(r):</b>	Carsten Kleiner
<b>Dozent(in):</b>	Carsten Kleiner, Frauke Sprengel
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      1. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium   = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Keine
<b>Angestrebte Lernergebnisse:</b>	Formale, algorithmische, mathematische Kompetenzen: formale Sprachen, die sie erzeugenden Grammatiken und die sie erkennenden Automaten kennen, Probleme mithilfe von Algorithmen durch entsprechenden Automaten lösen lassen, reguläre Ausdrücke kennen und für bestimmte Muster definieren können Analyse- und Design-Kompetenzen: Konzept der Berechenbarkeit kennen und nicht berechenbare Probleme erkennen, eigene Sprachen definieren und einen Parser dafür implementieren, Probleme analysieren, abstrahieren und die essentiellen Bestandteile mithilfe von einfachen Automaten lösen Methodenkompetenz: Übertragbarkeit von Problemen in verschiedene Lösungsvarianten erkennen können, alternative Beschreibungen derselben Sprachklasse erkennen und verwenden können
<b>Inhalt:</b>	Grundlegende Kenntnisse über Automaten- und Maschinenmodelle unterschiedlicher Komplexität (endliche Automaten, Kellerautomaten, Turing-Maschinen), verschiedene Klassen formaler Sprachen, Chomsky-Hierarchie und verschiedene Beschreibungsformen der Sprachen einer Klasse, Grundlagen des Compilerbaus
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion, Verwendung von Lehrsoftware Übung: Selbständige Bearbeitung der Aufgaben in der Übung, Abgabe von Hausaufgaben in 2er-Gruppen, Gemeinsame Begutachtung der vorgeschlagenen Lösungen sowie der Ideallösung an der Tafel
<b>Literatur:</b>	G. Vossen/K.-U. Witt: Grundkurs Theoretische Informatik, 3. Auflage, Vieweg J.Hopcroft/R. Motwani/J. Ullman: Introduction to Automata Theory, Languages and Computation (2.Auflage), Addison-Wesley



<b>Modulbezeichnung:</b>	<b>Datenstrukturen &amp; Algorithmen</b>
<b>Kürzel:</b>	DA
<b>Studiensemester:</b>	2. Semester (SS), Turnus: jährlich
<b>Modulverantwortliche(r):</b>	Carsten Kleiner
<b>Dozent(in):</b>	Ingo Ginkel, Matthias Hovestadt, Carsten Kleiner, Holger Peine, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      2. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium   = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Programmieren I
<b>Angestrebte Lernergebnisse:</b>	Formale und algorithmische Kompetenzen: Aufwandsabschätzungen mit Hilfe der O-Notation kennen und diese auf Algorithmen anwenden, Algorithmen bezüglich deren erwarteter Laufzeit vergleichen können, gute Algorithmen für wichtige Standardprobleme (z.B. Sortieren) kennen und vergleichen Analyse-, Design- und Realisierungs-Kompetenzen: wichtige lineare und nicht-lineare Datenstrukturen kennen und sinnvoll für ein gegebenes Problem einsetzen können, verschiedene Implementierungsvarianten für wichtige Datenstrukturen kennen, vergleichen und auswählen können, effiziente Datenstrukturen und Algorithmen für neue Probleme entwickeln können, Datenstrukturen aus Standardbibliotheken kennen und effizient einsetzen können
<b>Inhalt:</b>	Grundlegende Kenntnisse zur strukturierten und effizienten Software-Entwicklung: Analyse der Effizienz von Algorithmen, Lineare und nicht-lineare Datenstrukturen (Listen, Bäume, Heaps), Sortieralgorithmen, Paradigmen effizienter Algorithmen
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion, Ausführung und Visualisierung mithilfe von Lehrsoftware Übung: Selbständige Bearbeitung der Aufgaben in der Übung, Programmieraufgaben (Verwendung von Standarddatenstrukturen, Implementierung eigener Algorithmen), Abgabe von Hausaufgaben in 2er-Gruppen, Gemeinsame Begutachtung der vorgeschlagenen Lösungen sowie der Ideallösung
<b>Literatur:</b>	M. Goodrich/R. Tamassia: Data Structures and Algorithms in Java, 4th edition, Wiley T.H. Cormen, C.E. Leiserson, R.L. Rivest; Introduction to Algorithms; MIT Press

<b>Modulbezeichnung:</b>	<b>Datenbanken</b>
<b>Kürzel:</b>	DB
<b>Studiensemester:</b>	2. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Elisabeth Dennert-Möller
<b>Dozent(in):</b>	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner, Arne Koschel, Stefan Wohlfeil
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      2. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium   = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	Programmieren I, Mathematik I
<b>Angestrebte Lernergebnisse:</b>	Analyse-Kompetenz: sich in einen Anwendungsbereich einarbeiten können, Anforderungen extrahieren können, eine komplexe Domäne erfassen, strukturieren und auf der Basis von ER-Diagrammen modellieren können, Design-Kompetenz: aus Anforderungen einen Datenbankenentwurf ableiten können Technologische Kompetenz: Datenbankentwurf als Prozess Übergreifend: soziale Kompetenzen (Teamarbeit), Transferkompetenz
<b>Inhalt:</b>	In Datenbanken werden wichtige Elemente und Konzepte wie Datenmodellierung und relationale Datenmodelle vorgestellt. Unter anderem werden folgende Themen behandelt: <ul style="list-style-type: none"> <li>- Erstellen eines Datenbankentwurfs und Umsetzung in ein Datenbankschema,</li> <li>- Datenmanipulation im Relationenmodell,</li> <li>- SQL,</li> <li>- Normalisierung und objektorientierte Datenmodelle.</li> <li>- Anhand des Datenbanksystems Oracle wird der Stoff praktisch geübt und vertieft.</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbstständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	R. Elmasri, S. Navathe; Grundlagen von Datenbanksystemen; Pearson Education A. Heuer, G. Saake; Datenbanken: Konzepte und Sprachen; mitp A. Kemper, A. Eickler; Datenbanksysteme; Oldenburg

<b>Modulbezeichnung:</b>	<b>Betriebssysteme / Unix</b>
<b>Kürzel:</b>	BS
<b>Studiensemester:</b>	3. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Josef von Helden
<b>Dozent(in):</b>	Josef von Helden, Matthias Hovestadt, Holger Peine, Frank Müller
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor Pflichtfach 3. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Grundlagen der Informatik, Rechnerstrukturen, Programmieren I und II
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: Architektur, Konzepte und Funktionsweise moderner Betriebssysteme, insbes. Prozesse und Prozessstrukturen, Dateien und Dateisysteme, Systemsicherheit sowie die wichtigsten Schnittstellen verstehen. Analyse- und Realisierungskompetenz: Problemstellungen der Systemprogrammierung analysieren und in Lösungen überführen; nützliche und typische Unix-/Linux-Kommandos kennen lernen. Übergreifend: soziale Kompetenzen (Team-Arbeit), Transferkompetenz
<b>Inhalt:</b>	Es werden grundlegende Betriebssystem- und Rechnerstrukturen vorgestellt, insbesondere die Interrupt-Steuerung. Die Schwerpunkte des Moduls liegen auf den Themenbereichen Dateiverwaltung und -systeme sowie Prozessverwaltung. Als Querschnittsthemen werden Aspekte der Systemsicherheit und der Systemadministration aufgegriffen. In den Übungen werden die in der Vorlesung behandelten Sachverhalte beispielsweise durch Shellprogrammierung und Systemprogrammierung in C/C++ veranschaulicht und vertieft. Anhand komplexer Vorgänge im System (z.B. Booten und Anmelden) wird das Zusammenspiel der vorgestellten Einzelkomponenten demonstriert.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mdl. Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Demonstration, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Helmut Herold: „Linux- Unix Grundlagen. Kommandos und Konzepte“, Addison-Wesley H. Herold: „Linux- Unix- Systemprogrammierung“, Addison-Wesley Silberschatz, Abraham; Peter Galvin, Greg Gagne: „Operating System Concepts“, John Wiley & Sons Inc. Tanenbaum, Andrew S.: „Moderne Betriebssysteme“, Pearson Studium

<b>Modulbezeichnung:</b>	<b>Programmieren I (Einführung in die Programmierung: Java)</b>
<b>Kürzel:</b>	PR1
<b>Studiensemester:</b>	1. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Ralf Bruns
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Andreas Holitschke, Carsten Kleiner, Holger Peine, Stefan Wohlfeil
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      1. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 68 h Übung            = 34 h Eigenstudium   = 138 h
<b>Kreditpunkte:</b>	8 CP (= 240 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	keine
<b>Angestrebte Lernergebnisse:</b>	Algorithmische Kompetenz: eine konkrete Problemstellung analysieren und algorithmisch lösen können, grundlegende Algorithmen und Datenstrukturen zur Lösung von Problemen einsetzen Realisierungs-Kompetenz: Beherrschung des objektorientierten und des imperativen Programmierparadigmas, Erstellen und Testen von Programmen unter Einsatz entsprechender Werkzeuge
<b>Inhalt:</b>	Einführung in die Grundlagen der objektorientierten Programmierung anhand der Programmiersprache Java, deren Sprachkonstrukte mit einigen wichtigen Bibliotheken vorgestellt werden. Viele praktische Beispielaufgaben vertiefen den Stoff. Behandelt werden u.a. <ul style="list-style-type: none"> <li>– Grundlagen der Programmierung – Problem, Algorithmus, Programm</li> <li>– Grundlagen der objektorientierten Programmierung – Pakete, Klassen, Objekte</li> <li>– Einfache und strukturierte Datentypen</li> <li>– Kontrollstrukturen</li> <li>– Ein-/Ausgabe</li> <li>– Behandlung von Ausnahmen</li> <li>– Abstraktion</li> <li>– Rekursion</li> <li>– Prinzipien objektorientierter Programmierung – Vererbung und Polymorphie</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen
<b>Literatur:</b>	Horstmann, C.S., Cornell, G.: Core Java 2 – Band 1, Addison-Wesley. Horstmann, C.S., Cornell, G.: Core Java 2 – Band 2, Addison-Wesley. Eckel, B.: Thinking in Java, Prentice Hall.

<b>Modulbezeichnung:</b>	<b>Programmieren II (Objektorientierte Programmierung)</b>
<b>Kürzel:</b>	PR2
<b>Studiensemester:</b>	2. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Jürgen Dunkel
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Andreas Holitschke, Holger Peine
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      2. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 68 h Übung            = 34 h Eigenstudium   = 138 h
<b>Kreditpunkte:</b>	8 CP (= 240 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	-
<b>Empfohlene Voraussetzungen:</b>	Programmieren I
<b>Angestrebte Lernergebnisse:</b>	<p>Algorithmische Kompetenz: eine konkrete Problemstellung analysieren und algorithmisch lösen können.</p> <p>Analyse-Kompetenz: ein informell dargestelltes Problem mithilfe einer Modellierungssprache (UML) (semi-)formal beschreiben können.</p> <p>Design-Kompetenz: UML-Modelle in ablauffähige Programme umsetzen und dabei Grundregeln der Softwarearchitektur berücksichtigen.</p> <p>Realisierungs-Kompetenz: die entwickelten Programme können unter Einsatz entsprechender Werkzeuge erstellt und getestet werden.</p>
<b>Inhalt:</b>	<p>Auf Basis der in „Programmieren I“ gelegten Grundlagen werden weitere Konzepte der objektorientierten Programmierung vorgestellt. Vertieft behandelt werden u.a.</p> <ul style="list-style-type: none"> <li>- Vererbung und Polymorphie,</li> <li>- Ausnahmebehandlung mittels Exceptions und die</li> <li>- Entwicklung graphischer Benutzungsoberflächen mithilfe von Java-Swing.</li> </ul> <p>Darüber hinaus werden die Grundlagen objektorientierter Analyse und Designs, sowie die Unified Modeling Language (UML) eingeführt. Im Einzelnen werden die Schritte zur Entwicklung von Klassendiagrammen sowie der Einsatz dynamischer Modelle (Sequenz- und Aktivitätsdiagramme) vorgestellt.</p>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen</p>
<b>Literatur:</b>	<p>B. Oestereich: Analyse und Design mit UML 2.1., Oldenbourg Verlag.</p> <p>G. Krüger, T. Stark: Handbuch der Java-Programmierung, Addison-Wesley.</p> <p>B. Eckel: Thinking in Java, Prentice Hall,</p>

<b>Modulbezeichnung:</b>	<b>Programmieren III</b>
<b>Kürzel:</b>	PR3
<b>Studiensemester:</b>	3. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Rene Klingenberg
<b>Dozent(in):</b>	Holger Peine
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor Pflichtfach 3. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung 34 h Übung 34 h Eigenstudium 82 h
<b>Kreditpunkte:</b>	5 CP (= 150h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	
<b>Empfohlene Voraussetzungen:</b>	Programmieren I und II
<b>Angestrebte Lernergebnisse:</b>	<ul style="list-style-type: none"> <li>- Beherrschung der Programmiersprache C und wesentlicher Aspekte von C++.</li> <li>- Kenntnisse der algorithmischen Programmierung durch Anwendungen von C/C++ auf Berechnungsverfahren der numerischen Mathematik</li> </ul>
<b>Inhalt:</b>	Es werden Sprachkonzepte, Sprachelemente, Bibliotheksfunktionen und Compileroptionen in Theorie und Praxis behandelt. Viele Beispielprogramme dienen der Vertiefung des Stoffes.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben Übung: selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen
<b>Literatur:</b>	Klingenberg: Volltextskript zur Lehrveranstaltung Kernighan, B.W. u. D.M. Ritchie: Programmieren in ANSI C. Hanser V. Stroustrup, B.: Die C++-Programmiersprache. Addison-Wesley V. Breyman, U.: C++: Einführung und professionelle Programmierung. Hanser V.

<b>Modulbezeichnung:</b>	<b>Programmierprojekt</b>
<b>Kürzel:</b>	PP
<b>Studiensemester:</b>	3. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Klingenberg
<b>Dozent(in):</b>	Andreas Holitschke, alle Lehrenden
<b>Sprache:</b>	Java, C/C++
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      3. Semester
<b>Lehrform/SWS:</b>	4 SWS Projekt
<b>Arbeitsaufwand:</b>	180 h Projektarbeit
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	
<b>Empfohlene Voraussetzungen:</b>	Java, C/C++
<b>Angestrebte Lernergebnisse:</b>	Design und Realisierungskompetenzen: erworbenes technisches Wissen zur Lösung einer Problemstellung einsetzen; Erstellen und Testen eines größeren Software-Programms im Team Projektmanagementkompetenz: Fähigkeit zur Organisation und Steuerung von Projekten Soziale Kompetenzen: Teamarbeit, Kommunikation, Einüben von Präsentationen
<b>Inhalt:</b>	Softwareentwicklung zu ausgewählten Themen. Einarbeitung anhand von Literaturstudien. Aufstellen eines Projektplanes. Design, Implementierung und Dokumentation des Programms.
<b>Studien-/Prüfungsleistungen:</b>	Ergebnispräsentation im Kolloquium (ED)
<b>Medienformen:</b>	
<b>Literatur:</b>	Projektspezifisch

<b>Modulbezeichnung:</b>	<b>Mathematik I</b>
<b>Kürzel:</b>	MAT1
<b>Studiensemester:</b>	1. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Elisabeth Dennert-Möller
<b>Dozent(in):</b>	Volker Ahlers, Ingo Ginkel, Elisabeth Dennert-Möller, Frauke Sprengel
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      1. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 68 h Übung            = 34 h Eigenstudium   = 138 h
<b>Kreditpunkte:</b>	8 CP (= 240 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	keine
<b>Angestrebte Lernergebnisse:</b>	Formale Kompetenz: Kenntnisse der Logik und Vertrautheit mit mathematischen Formalismen zur Beschreibung von Sachverhalten Algorithmische Kompetenz: Kennenlernen mathematischer Algorithmen und ihrer Komplexität Mathematische Kompetenz: geeignete Lösungsverfahren für elementare Probleme der Mathematik und Informatik auswählen und durchführen Übergreifend: kommunikative Kompetenz (Präsentation und Diskussion von Lösungsvorschlägen)
<b>Inhalt:</b>	Die vermittelten Grundlagen der höheren Mathematik umfassen ausgewählte Themen aus den Bereichen <ul style="list-style-type: none"> <li>- Logik, Boolesche Algebra</li> <li>- Mengenlehre</li> <li>- Zahlensysteme</li> <li>- Funktionen und Relationen</li> <li>- Graphentheorie</li> <li>- lineare Algebra</li> <li>- elementare Zahlentheorie</li> </ul> Zur Veranschaulichung der Begriffe und Verfahren wird entsprechende Standardsoftware eingesetzt.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und Bearbeitung von Aufgaben als experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbstständige Bearbeitung der Aufgaben, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript zur Vorlesung Teschl, G., Teschl, S.: Mathematik für Informatiker, Springer - Verlag Hartmann, P.: Mathematik für Informatiker, Vieweg - Verlag



<b>Modulbezeichnung:</b>	<b>Mathematik II</b>
<b>Kürzel:</b>	MAT2
<b>Studiensemester:</b>	2. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Frauke Sprengel
<b>Dozent(in):</b>	Volker Ahlers, Ingo Ginkel, Elisabeth Dennert-Möller, Frauke Sprengel
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      2. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 68 h Übung          = 34 h Eigenstudium = 138 h
<b>Kreditpunkte:</b>	8 CP (= 240 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	Mathematik I
<b>Angestrebte Lernergebnisse:</b>	Formale und mathematische Kompetenz: Begriffe und Verfahren der Analysis kennen, die für das Verständnis der angewandten Verfahren in den Grundlagen der Numerik, Computergraphik, Bildverarbeitung und Animation erforderlich sind, Verfahren selbständig auch in anderen Bereichen der angewandten Informatik einsetzen können.
<b>Inhalt:</b>	Ausgewählte Themen aus den Bereichen Folgen und Reihen, Funktionen, Differentialrechnung in einer und mehreren Variablen, Integralrechnung einer Variablen, Funktionenreihen, gewöhnliche Differentialgleichungen, Anwendungsbeispiele, Veranschaulichung der Begriffe und Verfahren mit entsprechender Standardsoftware.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige oder gruppenweise Bearbeitung der Aufgaben, Begutachtung und Besprechung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript zur Vorlesung Teschl/Teschl: Mathematik für Informatiker 1/2, Springer, neueste Auflage Peter Hartmann: Mathematik für Informatiker, Vieweg, neueste Auflage Manfred Brill: Mathematik für Informatiker, Hanser Fachbuch, neueste Auflage

<b>Modulbezeichnung:</b>	<b>Mathematik III</b>
<b>Kürzel:</b>	MAT3
<b>Studiensemester:</b>	3. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Elisabeth Dennert-Möller
<b>Dozent(in):</b>	Volker Ahlers, Ingo Ginkel, Elisabeth Dennert-Möller, Frauke Sprengel
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      3. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 68 h Übung            = 34 h Eigenstudium   = 138 h
<b>Kreditpunkte:</b>	8 CP (= 240 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	keine
<b>Empfohlene Voraussetzungen:</b>	keine
<b>Angestrebte Lernergebnisse:</b>	Algorithmische und mathematische Kompetenz: Kennenlernen, Einsetzen, Vergleichen und Bewerten numerischer Verfahren, Auswahl und Einsatz geeigneter Verfahren für die näherungsweise Lösung von Problemen der Mathematik und Technik, Kennenlernen stochastischer Methoden zur Beschreibung und Analyse von Datenmengen Interpretation und Bewertung von Ergebnissen numerischer und stochastischer Verfahren Übergreifend: kommunikative Kompetenz (Präsentation und Diskussion von Lösungsvorschlägen)
<b>Inhalt:</b>	Die grundlegenden Begriffe der numerischen Verfahren und ihrer Fehler werden anhand ausgewählter Themen aus den Bereichen lineare Gleichungssysteme, Approximationstheorie, Quadraturverfahren und numerische Behandlung von Differentialgleichungen vermittelt. Die Grundlagen der Statistik beschäftigen sich mit - Kombinatorik - wahrscheinlichkeitstheoretischen Grundlagen - statistischer Datenanalyse - Regressions- und Korrelationsverfahren, Stochastische Prozesse Für beide Teilgebiete der angewandten Mathematik wird entsprechende Standardsoftware vorgestellt und eingesetzt.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbstständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript zur Vorlesung Hübner, G.: Stochastik, Vieweg - Verlag Locher, F.: Numerische Mathematik für Informatiker, Springer - Verlag Schwarz, H.R.: Numerische Mathematik. Teubner Verlag Teschl, G., Teschl, S.: Mathematik für Informatiker, Springer - Verlag Zurmühl, R.: Praktische Mathematik. Springer-Verlag

<b>Modulbezeichnung:</b>	<b>Grundlagen der Betriebswirtschaftslehre</b>
<b>Kürzel:</b>	BWL
<b>Studiensemester:</b>	3. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Josef von Helden
<b>Dozent(in):</b>	Lehrbeauftragte und Lehrende der Abteilung Wirtschaft
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor Pflichtfach 3. Semester
<b>Lehrform/SWS:</b>	4 SWS Vorlesung mit ca. 80 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung = 68 h Eigenstudium = 52 h
<b>Kreditpunkte:</b>	4 CP (= 120 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Keine
<b>Angestrebte Lernergebnisse:</b>	Fachübergreifende Kompetenz: Die Studierenden erwerben Wissen um betriebswirtschaftliche Zusammenhänge in einem beruflichen Umfeld. Die Studierenden kennen die verschiedenen Rechtsformen der Betriebe, sind mit der Kosten- und Leistungsrechnung vertraut, verfügen über grundlegende Kenntnisse in den Bereichen Personalwesen, Marketing und Controlling und lernen, wie industrielle Projekte unter marktwirtschaftlichen Gesichtspunkten realisiert werden.
<b>Inhalt:</b>	Grundzüge der Betriebsorganisation, Grundlagen des Controlling, Vermittlung der Grundlagen des betrieblichen Personalwesens und der Personalführung unter Berücksichtigung von Führungsstrukturen, Aufbau- und ablauforientierte Projektstrukturen, Aufwandsschätzung, Kapazitäts-, Termin-, Kosten- und Risikoplanung, Dokumentation, Projektsteuerung und -kontrolle, Projektleitung und Projektteam, Einsatz moderner Softwaretools.
<b>Studien-/Prüfungsleistungen:</b>	Benotete Prüfungsleistung
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Demonstration, Diskussion
<b>Literatur:</b>	Claus Steinle / Heike Bruch (Hrsg.), Controlling, Kompendium für Ausbildung und Praxis, Schäffer-Poeschel Verlag Stuttgart

<b>Modulbezeichnung:</b>	<b>Ergänzende Fächer I</b>
<b>Kürzel:</b>	EF1
<b>Lehrveranstaltungen:</b>	1. Ergänzendes Fach: EF1-1 2. Ergänzendes Fach: EF1-2
<b>Studiensemester:</b>	
<b>Modulverantwortliche(r):</b>	Klingenberg
<b>Dozent(in):</b>	Lehrbeauftragte
<b>Sprache:</b>	Deutsch, Englisch o.ä.
<b>Zuordnung zum Curriculum:</b>	EF1-1: Bachelor Pflichtfach 1. Semester EF1-2: Bachelor Pflichtfach 3. Semester
<b>Lehrform/SWS:</b>	EF1-1: 2 SWS Vorlesung mit ca. 80 Studierenden EF1-2: 2 SWS Vorlesung mit ca. 80 Studierenden
<b>Arbeitsaufwand:</b>	EF1-1: Vorlesung 34 h Eigenstudium 26 h EF1-2: Vorlesung 34 h Eigenstudium 26 h
<b>Kreditpunkte:</b>	EF1-1: 2 CP (= 60 h) EF1-2: 2 CP (= 60 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	
<b>Empfohlene Voraussetzungen:</b>	
<b>Angestrebte Lernergebnisse:</b>	EF1-1: Basiskompetenz im Englischen bezogen auf die durchschnittlichen Erfordernisse in der Berufsausübung eines Bachelors der Informatik EF1-2: Fachübergreifende Kompetenzen in einem studienübergreifenden Thema
<b>Inhalt:</b>	EF1-1: Englisch Erweiterung des grundlegenden Fachvokabulars und des allgemeinen Wortschatzes. Erwerb von Kenntnissen bei der Erstellung englischsprachiger fachbezogener Texte. Diskussions- und Vortragsübungen. Fachbezogenes Referieren. EF1-2: Gemäß Angebot Studienübergreifendes Thema aus den Gebieten wie z.B. Informationsrecht, Datenschutz, Moderationstechniken, Projektorganisation, Sprachen.
<b>Studien-/Prüfungsleistungen:</b>	EF1-1: Benotete Prüfungsleistung EF1-2: Benotete Prüfungsleistung
<b>Medienformen:</b>	Präsentation, Tafel, Beispielaufgaben, Diskussion
<b>Literatur:</b>	Fachspezifisch

<b>Modulbezeichnung:</b>	<b>Software Engineering I</b>
<b>Kürzel:</b>	SE1
<b>Studiensemester:</b>	4. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Ralf Bruns
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Arne Koschel, Holger Peine
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      4. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. Semesters
<b>Empfohlene Voraussetzungen:</b>	Programmieren I-III, Datenbanken, Programmierprojekt
<b>Angestrebte Lernergebnisse:</b>	Analyse-Kompetenz: sich in einen Anwendungsbereich einarbeiten können, Anforderungen extrahieren können, eine komplexe Domäne erfassen, strukturieren und modellieren können, Kenntnis von Standardsituationen im Bereich der Modellierung (Muster, Architekturen) haben Design-Kompetenz: aus Anforderungen einen Systementwurf ableiten können (aus der Kenntnis von Standardarchitekturen), einen Systementwurf in eine produktiv einsetzbare Implementierung überführen können, Qualitätssicherung als integralen Bestandteil des Entwicklungsprojektes verstehen. Übergreifend: Projekte aufsetzen und durchführen, soziale Kompetenzen (Teamarbeit), Transferkompetenz, eigenständiges Erarbeiten von neuen Methoden
<b>Inhalt:</b>	Vorgehensmodelle, Prozesse und Methoden zur Entwicklung großer Softwaresysteme, Entwurfsmuster (Patterns). Für einen Standard-Software-Entwicklungsprozess werden die verschiedenen Phasen und Arbeitsschritte im Detail vorgestellt. Im Einzelnen werden - für jede Phase der Softwareentwicklung die Grundlagen, die Ergebnissen und das Vorgehen behandelt, - Modellierung (mit UML) im Projektkontext eingeübt, - Analyse-, Design- und Architekturmuster eingeführt, - grundlegende Verfahren zur Qualitätssicherung vorgestellt.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Zuser, W., Grechenig, T., Köhle, M.: Software Engineering mit UML und dem Unified Process, Pearson. Jacobson, I., G. Booch, J. Rumbaugh: The Unified Software Development Process, Addison Wesley. Gamma, E. R., Helm, R. Johnson, J. Vlissides: Design Patterns, Reading, MA, Addison Wesley.

<b>Modulbezeichnung:</b>	<b>Software Engineering II</b>
<b>Kürzel:</b>	SE2
<b>Studiensemester:</b>	5. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Ralf Bruns
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Arne Koschel, Holger Peine
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      5. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. und 2. Semesters
<b>Empfohlene Voraussetzungen:</b>	Software Engineering I
<b>Angestrebte Lernergebnisse:</b>	<p>Projektmanagement-Kompetenz: Projekte planen, kontrollieren/ steuern können; ökonomische Randbedingungen und deren Auswirkungen kennen und beurteilen; Menschenführung als Erfolgsfaktor begreifen</p> <p>Design-Kompetenz: einen Systementwurf in eine produktiv einsetzbare Implementierung überführen können, Qualitätssicherung und Software-Ergonomie als integrale Bestandteile des Entwicklungsprojektes verstehen.</p> <p>Technologische Kompetenz: fundierte Kenntnisse im ausgewählten Themengebiet der Softwaretechnik.</p> <p>Übergreifend: Projekte aufsetzen und durchführen, soziale Kompetenzen (Teamarbeit), Transferkompetenz, eigenständiges Erarbeiten von neuen Methoden, Konfliktmanagement</p>
<b>Inhalt:</b>	<p>Projektmanagement, Qualitätsmanagement, Software-Ergonomie, Mensch-Maschine-Schnittstelle, weitere ausgewählte Themen aus dem Bereich Softwaretechnik.</p> <p>Vorgesehen als weitere ausgewählte Themen sind z.B. Themen aus einem oder mehreren der folgenden Gebiete</p> <ul style="list-style-type: none"> <li>- Internet-Technologien, insbesondere XML-basierte Informationssysteme</li> <li>- Vertiefung Mensch-Maschine-Schnittstelle</li> <li>- Auswirkungen der Informatik auf die Gesellschaft</li> <li>- Spezielle Programmiertechniken/-paradigmen</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung der Aufgaben, Vorstellung/ Diskussion von Lösungen durch Kurzvorträge Begutachtung der Lösungen, Gesprächsführung</p>
<b>Literatur:</b>	<p>Zuser, W., Grechenig, T., Köhle, M.: Software Engineering mit UML und dem Unified Process, Pearson.</p> <p>DeMarco, T, Lister, T.: Wien wartet auf Dich! Der Faktor Mensch im DV-Management, Hanser-Verlag.</p> <p>Balzert, H.: Lehrbuch der Software-Technik, Spektrum Akade. Verlag</p>

<b>Modulbezeichnung:</b>	<b>Software Engineering III</b>
<b>Kürzel:</b>	SE3
<b>Studiensemester:</b>	6. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Jürgen Dunkel
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Arne Koschel, Holger Peine
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      6. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium   = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1., 2. und 3. Semesters
<b>Empfohlene Voraussetzungen:</b>	Software Engineering I-II, Datenbanken
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: Verständnis für die Funktionsweise von verteilten Systemen insbesondere von webbasierten Anwendungen besitzen  Design-Kompetenz: grundlegende Konstruktionsprinzipien komplexer Softwarearchitekturen kennen und anwenden können. Einsatz geeigneter Entwurfsmuster zur Entwicklung verteilter Systeme.  Realisierungs-Kompetenz: technisch komplexe Infrastrukturen konfigurieren und einsetzen können.
<b>Inhalt:</b>	Grundlegende Konzepte von Softwarearchitekturen, insbesondere zur Entwicklung verteilter Systeme. <ul style="list-style-type: none"> <li>- Anhand eines einfachen Persistenz-Frameworks wird die Entwicklung von Frameworks vorgestellt.</li> <li>- Grundkonzepte verteilte Systeme, verteilte Softwarearchitekturen, Mehrschichtenarchitektur, Middleware</li> <li>- Implementierung verteilter Systeme: Sockets, RMI, MoM, Web Services</li> <li>- Architektur und Entwicklung internet-basierter Systeme: Servlets, JSPs, ..</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen,
<b>Literatur:</b>	J. Dunkel, A. Holitschke: Softwarearchitektur für die Praxis, Springer Verlag G. Bengel: Grundkurs verteilte Systeme, vieweg M. Boger: Java in verteilten Systemen, dpunkt.verlag A. Eberhart, S. Fischer: Web Services, Hanser Verlag

<b>Modulbezeichnung:</b>	<b>Informationssysteme I</b>
<b>Kürzel:</b>	IS1
<b>Studiensemester:</b>	4. Semester (SS), Turnus: jährlich
<b>Modulverantwortliche(r):</b>	Carsten Kleiner
<b>Dozent(in):</b>	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      4. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. Semesters
<b>Empfohlene Voraussetzungen:</b>	Programmieren I und II, Datenstrukturen&Algorithmen, Datenbanken, Betriebssysteme/UNIX
<b>Angestrebte Lernergebnisse:</b>	Formale und algorithmische Kompetenzen: Modellierungstechniken von Daten für ein Datenbanksystem kennen und implementieren können, Verarbeitung und Optimierung von Anfragen an ein Datenbanksystem kennen und beurteilen können, Grundprinzipien der Transaktionsverarbeitung kennen und in Szenarien anwenden können Analyse-, Design und Realisierungskompetenzen: Prinzipien zur automatisierten Datenwiederherstellung sowie zur Optimierung von Datenbanksystemen kennen, anwenden und realisieren können, Synchronisationsprobleme im Mehrbenutzerbetrieb erkennen, Lösungen finden und realisieren können Technologische Kompetenzen: interne Organisation und Speicherverwaltung eines Datenbanksystems kennen, Methoden zur Sicherung der Integrität von Transaktionen und Datenbankinhalten kennen und verwenden können, Normalformen für Schemata kennen
<b>Inhalt:</b>	Datenmodellierung und Normalformen, Integritätssicherung, Interne Datenbank- und Speicherorganisation, Verarbeitung und Optimierung von Anfragen, Transaktionsmanagement, Synchronisationsprobleme, Recovery und Tuning von Datenbanksystemen
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Diskussion, Visualisierungssoftware Übung: Selbständige Bearbeitung der Aufgaben in der Übung, Abgabe von Hausaufgaben in 2er-Gruppen, Gemeinsame Begutachtung der vorgeschlagenen Lösungen sowie der Ideallösung, Programmieraufgaben und/oder Projektaufgabe
<b>Literatur:</b>	R. Elmasri, S. Navathe: Grundlagen von Datenbanksystemen, Pearson Studium T. Härder, E. Rahm: Datenbanksysteme. Konzepte und Techniken der Implementierung, Springer



<b>Modulbezeichnung:</b>	<b>Informationssysteme II</b>
<b>Kürzel:</b>	IS2
<b>Studiensemester:</b>	5. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Arne Koschel
<b>Dozent(in):</b>	Felix Heine, Arne Koschel
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      5. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 60-80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium   = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. und 2. Semesters
<b>Empfohlene Voraussetzungen:</b>	Datenbanken, Informationssysteme I, Programmierprojekt, Programmieren I-III
<b>Angestrebte Lernergebnisse:</b>	Analyse-Kompetenz: Sich in einen Anwendungsbereich bzgl. Datenzugriff, Anforderungen extrahieren können, DB-Zugriffstechnologien anwendungsbezogen vergleichen, bewerten und auswählen können, Kenntnis von Standardsituationen im Bereich der Datenzugriffe (Muster, Web- und Enterprise-Architekturen) haben. Methodenkompetenz: Transferkompetenz für bestehende historisch gewachsene Datenzugriffsverfahren neue informatische Methoden in eine oft historisch gewachsene IT-Infrastruktur einzuführen. Technologische Kompetenzen: Verständnis für die Konzepte und Funktionsweise von: DB-Programmierung, Informationssysteme, Client-DB-Server-Strukturen, N-Tier Web-Architekturen
<b>Inhalt:</b>	Datenzugriffe und Datenverwaltung in Software- bzw. Informationssystemen – Konzepte, Technologien, Architekturen, Bewertung. Behandelte Themen beinhalten u.a.: <ul style="list-style-type: none"> <li>• DB-interne Programmierung (Stored Procedures, Trigger)</li> <li>• Relationale DB-Integration (statisch, dynamisch) auf Client-DB-Server-Basis.</li> <li>• Persistenz-Frameworks / O/R-Mapping</li> <li>• Fortgeschrittene Konzepte und Architekturen wie z.B. Java oder .NET Application Server, N-Tier-Web-Architekturen und objektrelationale DBMS.</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Conrad, S. et al.: „Enterprise Application Integration - Grundlagen - Konzepte - Entwurfsmuster - Praxisbeispiele. Spektrum/Elsevier Dunkel, A. Holitschke: „Softwarearchitektur für die Praxis“, Springer Heuer, A., Saake G., Sattler, K.: „Datenbanken: Konzepte und Sprachen“, mitp Kudraß, T. (Hrsg.): Taschenbuch Datenbanken, Hanser Aktuelle Web-Quellen, z.B. zu Hibernate, Java EE, OR-DBMS

<b>Modulbezeichnung:</b>	<b>Betriebssysteme und Netze I</b>
<b>Kürzel:</b>	BSN1
<b>Studiensemester:</b>	4. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Josef von Helden
<b>Dozent(in):</b>	Josef von Helden, Frank Müller, Matthias Hovestadt, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor Pflichtfach 4. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. Semesters
<b>Empfohlene Voraussetzungen:</b>	Betriebssysteme / Unix
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: Aufbau, Struktur und Technologie moderner, IP-basierter Netze kennen; Kommunikationsvorgänge in modernen Netzen nachvollziehen können. Analyse-, Methoden- und Realisierungskompetenz: Schnittstellen der Netzwerkprogrammierung anwenden, Netze konzipieren, aufbauen und konfigurieren, Methoden und Werkzeuge zum Lösen Netzprobleme selbstständig analysieren können. Übergreifend: soziale Kompetenzen (Team-Arbeit), Transferkompetenz
<b>Inhalt:</b>	Referenzmodelle; passive und aktive Netzwerkkomponenten z.B. Kabeltypen, Switches, Bridges, Hubs und Router; Protokolle verschiedener Schichten am Beispiel moderner IP-basierter Netzwerke, z.B. IEEE 802, IP, UDP und TCP; Switching und Routing. Praktische Übungen zu Netzwerkprogrammierung, IP-Konfiguration, IP-Adressierung, Routing und Protokollanalyse.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mdl. Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Demonstration, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, teilweise in Zusammenarbeit mit anderen Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	A. Sikora; Technische Grundlagen der Rechnerkommunikation, Fachbuchverlag Leipzig im Carl Hanser Verlag RRZN: „Netzwerke, Grundlagen“ D.Comer; TCP/IP, Konzepte, Protokolle Architekturen, mitp A. Tanenbaum; Computernetzwerke; Pearson Studium Shivendra S. Panwar, e.a.; TCP/IP Essentials, A Lab-Based Approach; Cambridge University Press W. R. Stevens; Unix Network Programming: The Sockets Networking API.: 1, Addison-Wesley

<b>Modulbezeichnung:</b>	<b>Betriebssysteme und Netze II</b>
<b>Kürzel:</b>	BSN2
<b>Studiensemester:</b>	5. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Stefan Wohlfeil
<b>Dozent(in):</b>	Stefan Wohlfeil, Josef von Helden, Matthias Hovestadt, Holger Peine
<b>Sprache:</b>	Deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor Pflichtfach 5. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. und 2. Semesters
<b>Empfohlene Voraussetzungen:</b>	Betriebssysteme/Unix und Betriebssysteme und Netze 1
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: Kenntnis der wichtigsten Anwendungsprotokolle, ihrer Funktionen und Eigenschaften auch im Hinblick auf Sicherheit. Kenntnis der (englischen) Fachbegriffe aus dem Bereich Betriebssysteme und Netze inklusive der genauen Bedeutung der Begriffe. Analyse- und Methoden-Kompetenz: Die Studenten können sich mit Hilfe von RFCs selbstständig in neue Protokolle einarbeiten, sie verstehen und die Einsatzmöglichkeiten abschätzen. Die Studenten können mit einfachsten Hilfsmitteln (einer Shell) die Anwendungsprotokolle benutzen und nach Problemen oder Fehlern suchen.
<b>Inhalt:</b>	Funktionsweise und Eigenschaften von wichtigen Anwendungsprotokollen in lokalen Netzen sowie im Internet: (DHCP, DNS, SMTP, POP3, IMAP4, FTP, HTTP, NFS, ...) Konzepte moderner Betriebssysteme (Verwaltung von CPU, Hauptspeicher, Festplatten, usw.); Einführung in IT-Sicherheit, Begriffe, Schutzziele, Übersicht über Kryptographie
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispiele, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	J. Kurose, K. Ross; <i>Computernetze</i> ; Pearson Education A. S. Tanenbaum: <i>Modern Operating Systems</i> ; Prentice Hall Press A. S. Tanenbaum: <i>Computernetzwerke</i> ; Pearson Studium A. Silberschatz; P. Galvin, P. Baer: <i>Operating System Concepts</i> , Addison-Wesley weitere Literatur je nach Schwerpunkt

<b>Modulbezeichnung:</b>	<b>Computergraphik I (Generative Computergraphik)</b>
<b>Kürzel:</b>	CG1
<b>Studiensemester:</b>	4. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Frauke Sprengel
<b>Dozent(in):</b>	Volker Ahlers, Frauke Sprengel
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      4. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. Semesters
<b>Empfohlene Voraussetzungen:</b>	Programmieren I-III, Mathematik I-II
<b>Angestrebte Lernergebnisse:</b>	Mathematische, algorithmische, fachübergreifende Kompetenz: kennen die mathematisch-geometrischen und physikalischen Grundlagen der Darstellungen und Beleuchtungen von Objekten im Raum und wenden diese Kenntnisse praktisch an. Analyse-, Design- und Realisierungskompetenz: wenden erworbene Fähigkeiten bei kleineren Projekten unter Verwendung einer Graphik-Bibliothek an.
<b>Inhalt:</b>	Grundbegriffe, Rastertechnik, affine und perspektive Abbildungen in der darstellenden Geometrie, Kurven und Flächen, Sichtbarkeit, Transparenz, Lokale Beleuchtungsmodelle, Einführung in eine Standard-3D-Bibliothek, Programmierbeispiele.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung praktischer Aufgaben in Kleingruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript zur Vorlesung Alan Watt: 3D-Computergraphik, Pearson Studium, neueste Auflage Foley, van Dam, Feiner, Hughes: Computer Graphics: Principles and Practice, Addison Wesley, neueste Auflage Alfred Nischwitz, Peter Haberäcker: Computergraphik und Bildverarbeitung, Vieweg, neueste Auflage

<b>Modulbezeichnung:</b>	<b>Computergrafik II (Bildverarbeitung)</b>
<b>Kürzel:</b>	CG2
<b>Studiensemester:</b>	5. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Elisabeth Dennert-Möller
<b>Dozent(in):</b>	Volker Ahlers, Elisabeth Dennert-Möller, Frauke Sprengel
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      5. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 80 Studierenden 2 SWS Übung mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Vorlesung      = 34 h Übung            = 34 h Eigenstudium   = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. und 2. Semesters
<b>Empfohlene Voraussetzungen:</b>	Mathematik 1 – 3, Computergrafik 1
<b>Angestrebte Lernergebnisse:</b>	Algorithmische und mathematische Kompetenz: Kennenlernen und Einsetzen von Verfahren der digitalen Bildverarbeitung, Analyse – und Design - Kompetenz: Auswahl und Einsatz geeigneter Verfahren für die Bildverbesserung in verschiedenen Anwendungsszenarien und ihre Realisierung in Hard- und Software Fachübergreifende Kompetenz durch theoretische und praktische Erfahrungen im Einsatz digitaler Bildverarbeitungsverfahren in verschiedenen Anwendungsbereichen soziale Kompetenzen durch Teamarbeit und Präsentation und Diskussion von Lösungsvorschlägen für ausgewählte Aufgaben
<b>Inhalt:</b>	Grundkenntnisse über Aufbau und Funktionsweise digitaler Bildverarbeitungssysteme mit praktischen Anwendungen: Grundbegriffe digitaler Bilder und ihrer Darstellung im Orts- und Frequenzraum Farbmodelle und Farbmanagement Verfahren zur Bildverbesserung im Orts- und im Frequenzraum Einfache Bildsegmentierungsverfahren Implementierungen von Bildverarbeitungsalgorithmen
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbstständige Bearbeitung von Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript zur Vorlesung Burger, W., Burge, M. J.: Digitale Bildverarbeitung, <a href="#">X.media.press</a> Springer - Verlag Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Prentice Hall Jähne, B. Digitale Bildverarbeitung, Springer - Verlag

<b>Modulbezeichnung:</b>	<b>Seminar</b>
<b>Kürzel</b>	BSEM
<b>Studiensemester:</b>	4. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Stefan Wohlfeil
<b>Dozent(in):</b>	Alle Dozentinnen und Dozenten
<b>Sprache:</b>	Deutsch (oder Englisch)
<b>Zuordnung zum Curriculum</b>	Bachelor Pflichtfach 4. Semester
<b>Lehrform/SWS:</b>	2 SWS Seminar mit ca. 20 Studierenden
<b>Arbeitsaufwand:</b>	Eigener Vortrag = 2 h; Teilnahme andere Vorträge = 32 h Eigenstudium = 66 h; Erstellung der Seminararbeit = 80 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Alle Modulprüfungen des 1. Semesters
<b>Empfohlene Voraussetzungen:</b>	Alle Bachelor Module der Semester eins bis drei.
<b>Angestrebte Lernergebnisse:</b>	<p>Fachliche Kompetenz: Im Seminar vertiefen die Studierenden exemplarisch Inhalte der vorherigen Veranstaltungen oder ergänzen sie. Durch die Einarbeitung in neue fachliche Aspekte der Informatik werden die Analyse-Kompetenzen verbessert während die technologischen Kompetenzen sich durch die Beschäftigung mit den neuen Aspekten entsprechend erweitern.</p> <p>Soziale Kompetenz und Selbstkompetenz: Die Studierenden vertiefen ihre Kenntnisse im selbstständigen Arbeiten mit wissenschaftlicher Literatur. Sie lernen, fachwissenschaftliche Inhalte überzeugend mündlich und schriftlich zu Präsentieren und verbessern somit ihre Methoden-Kompetenz. In Diskussionen wird die Fähigkeit zur kritischen Reflektion geübt.</p>
<b>Inhalt:</b>	Die Teilnehmer erarbeiten selbstständig ein anspruchsvolles Thema, erstellen einen schriftlichen Bericht und präsentieren die Ergebnisse. Dabei werden wissenschaftliche Methoden und Techniken angewendet.
<b>Studien-/Prüfungsleistungen:</b>	Schriftliche Ausarbeitung, eigener Vortrag, aktive Teilnahme an den anderen Vorträgen, Beteiligung an der Diskussion.
<b>Medienformen:</b>	
<b>Literatur:</b>	Wird zu Beginn der Lehrveranstaltung von den Dozenten bekannt gegeben.

<b>Modulbezeichnung:</b>	<b>Wahlpflichtfach Informatik</b>
<b>Kürzel:</b>	WPI
<b>Lehrveranstaltungen:</b>	Schwerpunkt 1: Technische Anwendungen Schwerpunkt 2: Computergraphik III (Animation)
<b>Studiensemester:</b>	6. Semester (SoSe), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Arne Koschel
<b>Dozent(in):</b>	Arne Koschel, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch
<b>Zuordnung Curriculum:</b>	Bachelor                      Pflichtfach                      6. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung mit ca. 40 Studierenden (pro Veranstaltung) 2 SWS Übung mit ca. 20 Studierenden In der Regel werden zwei thematisch unterschiedliche Veranstaltungen angeboten, aus denen die Studierenden eine auswählen müssen.
<b>Arbeitsaufwand:</b>	Vorlesung            = 34 h Übung                = 34 h Eigenstudium       = 82 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. – 3. Semesters.
<b>Empfohlene Voraussetzungen:</b>	Programmierprojekt, Programmieren I-III, Datenbanken, Betriebssysteme und Netze I, II, Computergraphik I, Mathematik III
<b>Angestrebte Lernergebnisse:</b>	Technologische und Realisierungskompetenzen wie folgt: Schwerpunkt 1: Verständnis für die Konzepte und Funktionsweise von weiterführenden Themen aus der Parallelverarbeitung und verteilten Systemen und deren Anwendung. Ausgewählte Werkzeuge und Konzepte zur professionellen Erstellung größerer Programmsysteme kennenlernen. Praktische Erprobung von Beispielarbeiten zu Themen wie Threading, GRID, MPI und P2P-Netze. Schwerpunkt 2: Anwendung der in den vorangehenden Semestern erworbenen Kenntnisse für eine praxisübliche Animationsaufgabe, u.a.: Möglichkeiten und Grenzen von OpenGL in der Animation; Kenntnisse der praktischen Anwendung von Fixpunktiterationen im $\mathbb{R}^3$ ; Kenntnisse in der Programmierung numerischer Lösungsverfahren nichtlinearer Gleichungssysteme; Praktische Anwendungen der linearen Algebra und Analysis im $\mathbb{R}^3$ .
<b>Inhalt:</b>	Schwerpunkt 1: Themenauswahl aus Konzepten und Werkzeugen für praktische Projektarbeiten einerseits und Aspekten des Hochleistungsrechnens und verteilter Systeme andererseits. Schwerpunkt 2: Animation der räumlichen Bewegung hierarchischer und auch geschlossener kinematischer Ketten (z.B. Roboter etc.) oder weitere Methoden der Computeranimation.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung), experimentelle Arbeit, ggf. auch praktische und theoretische Ausarbeitungen.
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, Erstellung & Dokumentation von Programmen
<b>Literatur:</b>	Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., Koschel, A. Systemarchitekturen für verteilte Anwendungen, Hanser. Oaks, S., Wong, H.: Java Threads, O'Reilly & Associates Klingenberg: Volltext-Skript zur Lehrveranstaltung Computergraphik III Woernle: Dynamik von Mehrkörpersystemen (Skript UNI-Rostock)

<b>Modulbezeichnung:</b>	<b>Ergänzende Fächer II</b>
<b>Kürzel:</b>	EF2
<b>Lehrveranstaltungen:</b>	3. Ergänzendes Fach: EF2-1 4. Ergänzendes Fach: EF2-2
<b>Studiensemester:</b>	4. Semester
<b>Modulverantwortliche(r):</b>	Klingenberg
<b>Dozent(in):</b>	I.d.R. Lehrbeauftragte
<b>Sprache:</b>	Beliebig
<b>Zuordnung zum Curriculum:</b>	EF2-1: Bachelor Pflichtfach 4. Semester EF1-2: Bachelor Pflichtfach 4. Semester
<b>Lehrform/SWS:</b>	EF2-1: 2 SWS Vorlesung mit ca. 80 Studierenden EF2-2: 2 SWS Vorlesung mit ca. 80 Studierenden
<b>Arbeitsaufwand:</b>	EF2-1: Vorlesung 34 h Eigenstudium 26 h EF2-2: Vorlesung 34 h Eigenstudium 26 h
<b>Kreditpunkte:</b>	EF1-1: 2 CP (= 60 h) EF1-2: 2 CP (= 60 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. Semesters
<b>Empfohlene Voraussetzungen:</b>	EF2-1: BWL
<b>Angestrebte Lernergebnisse:</b>	EF2-1: Fachübergreifende Kompetenz in einem betriebswirtschaftlichen Thema EF2-2: Fachübergreifende Kompetenzen in einem studienübergreifenden Thema
<b>Inhalt:</b>	EF2-1: Frei wählbares Fach mit Themen aus der Betriebswirtschaftslehre EF2-2: Studienübergreifendes Thema aus Gebieten wie z.B. Informationsrecht, Datenschutz, Moderationstechniken, Projektorganisation, Sprachen, wissenschaftliche Datenverarbeitung.
<b>Studien-/Prüfungsleistungen:</b>	EF2-1: Benotete Prüfungsleistung EF2-2: Benotete Prüfungsleistung
<b>Medienformen:</b>	Präsentation, Tafel, Beispielaufgaben, Diskussion
<b>Literatur:</b>	Fachspezifisch



<b>Modulbezeichnung:</b>	<b>Praxisprojekt I</b>
<b>Kürzel:</b>	BPR1
<b>Studiensemester:</b>	5. Semester (WS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Ralf Bruns
<b>Dozent(in):</b>	Alle Lehrenden
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      5. Semester
<b>Lehrform/SWS:</b>	8 SWS Projekt mit ca. 13 Studierenden
<b>Arbeitsaufwand:</b>	Projektarbeit = 300 h
<b>Kreditpunkte:</b>	10 CP (= 300 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1. und 2. Semesters
<b>Empfohlene Voraussetzungen:</b>	Projektspezifisch
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse-Kompetenzen: eine gegebene Fragestellung unter Einsatz wissenschaftlicher Methoden und Erkenntnisse eigenständig analysieren können, sich in Aufgabenstellungen verschiedener Anwendungsfelder einarbeiten können.</p> <p>Design- /Realisierungs-Kompetenzen: Lösungsstrategien entwickeln und diese mit den erworbenen Kenntnissen umzusetzen können.</p> <p>Technologische Kompetenzen: Wissen aus verschiedenen Bereichen kombinieren und zielgerichtet einsetzen können.</p> <p>Methoden-Kompetenzen: verschiedene innovative Methoden zur Lösung praktischer Probleme anwenden können.</p> <p>Projektmanagement-Kompetenz: Fähigkeit zur Projektplanung, zum Aufbau einer Organisationsstruktur und zur Steuerung von Projekten.</p> <p>Soziale Kompetenzen: Anwendung von Konfliktlösungsstrategien, Teamarbeit, Kommunikation</p>
<b>Inhalt:</b>	<p>Bearbeitung einer komplexen Aufgabenstellung aus der Praxis in einer Projektgruppe. Kennzeichnend ist die projektspezifische Organisation, Planung und Durchführung. Das Projekt erstreckt sich in der Regel über einen Zeitraum von 2 Semestern.</p> <p>Die Abteilung Informatik strebt an, die Praxisprojekte (wie bisher) in Kooperation mit externen Partnern, meistens Industrieunternehmen, anzubieten, da i.d.R. betriebswirtschaftlich oder industriell relevante Fragestellungen bearbeitet werden. Die Aufgabenstellungen stammen demzufolge aus verschiedenen Anwendungsfeldern.</p>
<b>Studien-/ Prüfungsleistungen:</b>	Experimentelle Arbeit
<b>Medienformen:</b>	Projektarbeit: Präsentation, Tafel, Beispielaufgaben, Diskussion Selbständige Bearbeitung der Aufgaben, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Projektspezifisch

<b>Modulbezeichnung:</b>	<b>Praxisprojekt II</b>
<b>Kürzel:</b>	BPR2
<b>Studiensemester:</b>	6. Semester (SS), Turnus: jedes Jahr
<b>Modulverantwortliche(r):</b>	Ralf Bruns
<b>Dozent(in):</b>	Alle Lehrenden
<b>Sprache:</b>	deutsch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      6. Semester
<b>Lehrform/SWS:</b>	4 SWS Projekt mit ca. 13 Studierenden
<b>Arbeitsaufwand:</b>	Projektarbeit = 150 h
<b>Kreditpunkte:</b>	5 CP (= 150 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Alle Modulprüfungen des 1., 2. und 3. Semesters
<b>Empfohlene Voraussetzungen:</b>	Projektspezifisch
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse-Kompetenzen: eine gegebene Fragestellung unter Einsatz wissenschaftlicher Methoden und Erkenntnisse eigenständig analysieren können, sich in Aufgabenstellungen verschiedener Anwendungsfelder einarbeiten können.</p> <p>Design- /Realisierungs-Kompetenzen: Lösungsstrategien entwickeln und diese mit den erworbenen Kenntnissen umzusetzen können.</p> <p>Technologische Kompetenzen: Wissen aus verschiedenen Bereichen kombinieren und zielgerichtet einsetzen können.</p> <p>Methoden-Kompetenzen: verschiedene innovative Methoden zur Lösung praktischer Probleme anwenden können.</p> <p>Projektmanagement-Kompetenz: Fähigkeit zur Projektplanung, zum Aufbau einer Organisationsstruktur und zur Steuerung von Projekten.</p> <p>Soziale Kompetenzen: Anwendung von Konfliktlösungsstrategien, Teamarbeit, Kommunikation</p>
<b>Inhalt:</b>	Fortsetzung des Moduls „Praxisprojekt I“ aus dem vorigen Semester. Bearbeitung einer komplexen Aufgabenstellung aus der Praxis in einer Projektgruppe. Kennzeichnend ist die projektspezifische Organisation, Planung und Durchführung.
<b>Studien-/ Prüfungsleistungen:</b>	Experimentelle Arbeit
<b>Medienformen:</b>	Projektarbeit: Präsentation, Tafel, Beispielaufgaben, Diskussion Selbständige Bearbeitung der Aufgaben, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Projektspezifisch

<b>Modulbezeichnung:</b>	<b>Bachelorarbeit mit Kolloquium</b>
<b>Kürzel:</b>	BA
<b>Studiensemester:</b>	6.Semester (SS); Turnus: halbjährlich
<b>Modulverantwortliche(r):</b>	Carsten Kleiner
<b>Dozent(in):</b>	Alle Lehrenden
<b>Sprache:</b>	Deutsch/Englisch
<b>Zuordnung zum Curriculum:</b>	Bachelor      Pflichtfach      6. Semester
<b>Lehrform/SWS:</b>	Abschlussarbeit mit Kolloquium, Einzelarbeit (Gruppenarbeit möglich), keine festen Präsenzzeiten
<b>Arbeitsaufwand:</b>	Eigenarbeit = 360 h Vorbereitung und Durchführung Kolloquium = 90 h
<b>Kreditpunkte:</b>	12 CP Bachelorarbeit + 3 CP Kolloquium (= 450 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Bestandene Vorprüfung, mind. 134 CP erworben aus anderen Modulen gemäß Studienhandbuch
<b>Empfohlene Voraussetzungen:</b>	Alle Module des 4.+5.Semesters aus dem gewählten Themenbereich, Seminar, Praxisprojekt I
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse-, Design- und Realisierungskompetenzen: einen Themenbereich aus der gewählten Fächergruppe selbständig analysieren und nach wissenschaftlichen Methoden bearbeiten können, Lösungen konzeptionieren und realisieren</p> <p>Technologische Kompetenzen: Technologien aus dem gewählten Themenbereich auswählen und zur Problemlösung einsetzen können</p> <p>Fachübergreifende Kompetenzen: das gewählte Thema soll in Zusammenarbeit mit einem Wirtschaftsunternehmen bearbeitet werden, die Rahmenbedingungen des Unternehmens sind in die Lösung einzubeziehen</p> <p>Methodenkompetenzen: Stand des Wissens zum gewählten Thema mit wissenschaftlichen Methoden erschließen können, bekannte Lösungen für das gegebene Thema anpassen und erweitern können</p> <p>Projektmanagement-Kompetenz: Themenstellung in einer vorgegebenen Zeit bearbeiten können, Planung der erforderlichen Teilaufgaben und Überwachung des Zeitplans, Selbstorganisation durchführen und Fertigstellungstermin einhalten</p> <p>Selbstkompetenz: gewähltes Thema sowie entwickelte Lösung fachgerecht präsentieren können, fachliche Fragen dazu beantworten können, alternative Lösungen diskutieren und einschätzen können</p>
<b>Inhalt:</b>	Selbständige Bearbeitung eines Themas aus der Angewandten Informatik nach wissenschaftlichen Methoden, Erstellung einer schriftlichen Ausarbeitung, Präsentation der Ergebnisse und kritische Diskussion des Themas; das Thema soll aus dem Bereich eines Wirtschaftsunternehmens stammen oder direkt dort bearbeitet werden
<b>Studien-/Prüfungsleistungen:</b>	Abschlussarbeit, Kolloquium
<b>Medienformen:</b>	Eigenständige wissenschaftliche Arbeit, Erstellung eines Dokuments, Vorbereitung und Durchführung einer Präsentation
<b>Literatur:</b>	Themenabhängig

## **Masterstudiengang Angewandte Informatik – Modulbeschreibungen**

Vorlesungsmodule im gemeinsamen Block werden mit allen Masterstudierenden durchgeführt, also bis zu ca. 36 Studierenden in den Vorlesungen und typischerweise zwei Übungsgruppen mit je bis zu ca. 18 Studierenden.

Die Vorlesungsmodule in den einzelnen Schwerpunkten werden mit entsprechend anteiligen Master-Studierendenzahlen durchgeführt, also bis zu ca. 10-20 Studierenden in den Vorlesungen und typischerweise je einer Übungsgruppe mit bis zu ca. 10-20 Studierenden.

## 1. Gemeinsamer Block

<b>Modulbezeichnung:</b>	<b>Softwarearchitekturen</b>
<b>Kürzel</b>	MIN-SWA
<b>Studiensemester:</b>	1. Semester (SoSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Jürgen Dunkel
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Arne Koschel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, Pflichtfach, 1. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Keine
<b>Angestrebte Lernergebnisse:</b>	<p>Technologische Kompetenz: es sollen tiefgehende Fachkenntnisse im Gebiet Softwarearchitekturen vermittelt werden. Dabei soll insbesondere ein umfassendes Verständnis für die Anwendbarkeit der vorgestellten Techniken und Methoden sowie deren Grenzen entwickelt werden.</p> <p>Methodenkompetenz: Anwendung und kritische Bewertung der vorgestellten innovativen Methoden für die Entwicklung komplexer Softwarearchitekturen.</p> <p>Analyse-/Design und Realisierungskompetenz: die Fähigkeit, Anforderungen sowie die Erfolgsfaktoren von Softwarearchitekturen definieren zu können; Verwendung formaler Methoden und Modellierungssprachen zur Beschreibung von Softwarearchitekturen; Einsatz von Architekturmodellen zur Umsetzung von Systemarchitekturen.</p>
<b>Inhalt:</b>	<p>Grundkonzepte und Eigenschaften moderner Softwarearchitekturen, Entwurfskriterien, fortgeschritten Techniken zur Entwicklung komplexer Softwaresysteme. Dabei sollen insbesondere aktuelle Themen und Entwicklungen berücksichtigt werden. fortgeschrittene Ansätze zur Modellierung und Realisierung komplexer Systeme, z.B. Design-By-Contract mit OCL, Domain-Driven-Design, Modellgetriebene Softwareentwicklung (Domain Specific Languages), Geschäftsprozessmodellierung.</p> <p>Forschungsnahe Themen können ggf. in Masterarbeiten fortgeführt werden.</p>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung</p>
<b>Literatur:</b>	Aktuell nach Lehrinhalten

<b>Modulbezeichnung:</b>	<b>Projekt- und Qualitätsmanagement</b>
<b>Kürzel:</b>	MIN-PMQM
<b>Studiensemester:</b>	1. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Arne Koschel
<b>Dozent(in):</b>	Ralf Bruns, Arne Koschel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, Pflichtfach, 1. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Abgeschlossenes Bachelorstudium mit Informatikanteil, der mindestens den Programmieren- und Softwareengineering-Veranstaltungen vergleichbar ist.
<b>Angestrebte Lernergebnisse:</b>	Analyse, Design- und Methodenkompetenz: Die Studierenden erwerben die Fähigkeit zu eigenverantwortlicher Tätigkeit im Berufsfeld. Sie sollen Methoden und Konzepte wesentlicher Aspekte des Projekt- und Qualitätsmanagement erlernen um auch als Leiter eines Teams, das aus unterschiedlichen Disziplinen und Niveaus bestehen kann, arbeiten zu können.
<b>Inhalt:</b>	Es werden grundlegende Organisationsformen und Abläufe zur Durchführung von Projekten vorgestellt. Dazu gehören u.a. <ul style="list-style-type: none"> <li>• Projekt-Akquisition,</li> <li>• Projektorganisation und -planung,</li> <li>• Maßnahmen zum Projektcontrolling und Risiko-Management,</li> <li>• Konfigurations-Management</li> <li>• Qualitätssicherungsmaßnahmen</li> </ul> Darüber hinaus werden Vorgehensmodelle zur Sicherstellung der Prozessqualität diskutiert, wie bspw. ISO 9000 oder CMM.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Seminarthemen, Ausarbeitung, Ergebnispräsentation, Begutachtung, experimentelle Arbeit, Gesprächsführung
<b>Literatur:</b>	DeMarco, T., Lister, T.: „Wien wartet auf Dich“, Hanser. DeMarco, T.: „Der Termin“, Hanser. Balzert, H.: „Lehrbuch der Software-Technik 1/2“. Spektrum. Cockburn, A.: „Agile Software-Entwicklung“, Mitp (bzw. Engl. Original). Beck, K., Andres, C.: „Extreme Programming Explained“ (und: Kent Beck: Extreme Programming). Weitere Quellen zu speziellen Themen.

<b>Modulbezeichnung:</b>	<b>Visualisierung und HCI</b>
<b>Kürzel:</b>	MIN-VISH
<b>Studiensemester:</b>	2. Semester (SoSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Volker Ahlers
<b>Dozent(in):</b>	Volker Ahlers, Ingo Ginkel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, Pflichtfach, 2. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung/Projekt
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Keine
<b>Angestrebte Lernergebnisse:</b>	<p>Algorithmische und mathematische Kompetenzen: Kenntnis der Grundlagen der Mensch-Computer-Interaktion (HCI); Verständnis von Visualisierungsalgorithmen; Verständnis unterschiedlicher Datenrepräsentationen</p> <p>Analyse-, Design- und Realisierungskompetenzen: Analyse von Datensätzen und Visualisierungsanforderungen; Design und Realisierung von Visualisierungslösungen, Implementierung von Algorithmen</p> <p>Technologische Kompetenzen: Kenntnis des Einsatzes von Visualisierungstechniken in unterschiedlichen Anwendungsgebieten</p> <p>Methodenkompetenzen: Kenntnis der Möglichkeiten, des Nutzens und der Grenzen des Einsatzes von Visualisierungstechniken, Erkennen von Fehlern in Visualisierungslösungen</p>
<b>Inhalt:</b>	<ul style="list-style-type: none"> <li>• Grundlagen: Mensch-Computer-Interaktion (HCI), Wahrnehmung und Kognition, Gestaltgesetze, Farbmodelle, Datenrepräsentation</li> <li>• Skalare Daten: Diagramme (z.B. Linien-, Säulen-, Streudiagramm, Histogramm), Farbkodierung, Konturlinien, multivariate Daten</li> <li>• Volumendaten: Isoflächen, Volume Rendering, Ray Casting</li> <li>• Vektorfelder: Glyphen, Stromlinien, Strombahnen</li> <li>• Beziehungen: Visualisierung von Hierarchien (Bäumen) und Netzwerken (Graphen), z.B. Tree Map, kräftebasiertes (force-directed) Layout</li> <li>• Software: Aktuelle Visualisierungsbibliotheken und grafische Entwicklungsumgebungen, Anwendung auf reale Daten</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung von Aufgaben in Gruppen, Projektarbeit mit Ergebnispräsentation, Begutachtung und Diskussion der Lösungen, Gesprächsführung</p>
<b>Literatur:</b>	<p>Skript zur Vorlesung.</p> <p>Telea, A.C.: Data Visualization. A K Peters, neueste Auflage</p> <p>Ward, M., Grinstein, G.G., Keim, D.: Interactive Data Visualization. A K Peters, neueste Auflage</p> <p>Ware, C.: Information Visualization. Morgan Kaufmann, neueste Auflage</p>

<b>Modulbezeichnung:</b>	<b>Algorithmen und Komplexität</b>
<b>Kürzel:</b>	MIN-AK
<b>Studiensemester:</b>	3. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Ingo Ginkel
<b>Dozent(in):</b>	Ingo Ginkel, Frauke Sprengel, Felix Heine, Carsten Kleiner, Robert Garmann
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, Pflichtfach, 3. Semester
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Datenstrukturen und Algorithmen, Theoretische Informatik aus dem Bachelor-Studium
<b>Angestrebte Lernergebnisse:</b>	Theoretische und mathematische Kompetenzen: Charakterisierung des Schwierigkeitsgrads vorgegebener Probleme sowie deren Klassifizierung bezüglich effizienter algorithmischer Lösbarkeit. Analyse-, Design- und Realisierungskompetenzen: Kenntnis verschiedener algorithmischer Lösungsstrategien zur Bearbeitung praxisrelevanter Probleme von hoher Komplexität (Zeit/Speicher) Technologische Kompetenzen: Implementierung dieser Algorithmen
<b>Inhalt:</b>	<ul style="list-style-type: none"> <li>Theorie: Berechenbarkeit, Klassen P und NP, NP-Vollständigkeit, Polynomielle Reduktion, Satz von Cook-Levin, Graphenalgorithmen</li> <li>Lösungsstrategien: Divide and Conquer (Master Theorem), Backtracking, Branch-and-Bound, lokale Verbesserung, lineare Programmierung, genetische Algorithmen, Simulated Annealing, Randomisierung, Bin-Packing</li> <li>Probleme: z.B. Max-Cut Problem, Travelling Salesman Problem, Rucksackproblem, Eulerkreise, Minimum Spanning Tree, schnelle Potenzierung</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung von Aufgaben in Zweiergruppen, Projektarbeit mit Ergebnispräsentation, Begutachtung und Diskussion der Lösungen, Gesprächsführung
<b>Literatur:</b>	Folien/Skript zur Vorlesung Hromkovic: Algorithmics for Hard Problems: Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics, Springer 2004 Wegener: Komplexitätstheorie - Grenzen der Effizienz von Algorithmen. Springer, 2003



<b>Modulbezeichnung:</b>	<b>Masterseminar</b>
<b>Kürzel</b>	MIN-MSEM
<b>Studiensemester:</b>	2. Semester (SoSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Stefan Wohlfeil
<b>Dozent(in):</b>	Alle Dozentinnen und Dozenten
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, Pflichtfach, 2. Semester
<b>Lehrform/SWS:</b>	2 SWS Seminar mit ca. 5-15 Studierenden
<b>Arbeitsaufwand:</b>	Eigener Vortrag = 2 h; Teilnahme an anderen Vorträgen = 28 h Eigenstudium = 70 h; Erstellung der Seminararbeit = 80 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	
<b>Empfohlene Voraussetzungen:</b>	Alle Master Module des ersten Semesters
<b>Angestrebte Lernergebnisse:</b>	<p>Fachliche Kompetenz: Im Seminar vertiefen die Studierenden exemplarisch die Inhalte der vorherigen Veranstaltungen und erweitern ihre analytischen Kompetenzen indem sie neue und in der Forschung/Entwicklung befindliche Themen bearbeiten. Da diese aktuellen Themen auch aus angrenzenden Fachgebieten stammen vertiefen die Studierenden ihre fachübergreifenden Sachkompetenzen.</p> <p>Methodenkompetenz: Die Studierenden vertiefen ihre Kenntnisse im selbständigen Arbeiten mit anspruchsvoller wissenschaftlicher Literatur.</p> <p>Soziale Kompetenz und Selbstkompetenz: Die Studierenden lernen, fachwissenschaftliche Inhalte überzeugend mündlich und schriftlich zu präsentieren. In Diskussionen wird die Fähigkeit zur kritischen Reflektion geübt. Durch die Pflicht zur Einhaltung von Terminen unter nur begrenzt zur Verfügung stehenden Ressourcen wird die Projektmanagementkompetenz geschult sowie die Lern- und Leistungsbereitschaft gefördert.</p>
<b>Inhalt:</b>	Die Teilnehmer erarbeiten selbstständig ein anspruchsvolles wissenschaftliches Thema, erstellen einen schriftlichen Bericht und präsentieren die Ergebnisse. Dabei werden wissenschaftliche Methoden und Techniken angewendet.
<b>Studien-/Prüfungsleistungen:</b>	Schriftliche Ausarbeitung, eigener Vortrag, aktive Teilnahme an den anderen Vorträgen, Beteiligung an der Diskussion.
<b>Medienformen:</b>	
<b>Literatur:</b>	Wird zu Beginn der Lehrveranstaltung von den Dozenten bekannt gegeben.

<b>Modulbezeichnung:</b>	<b>Masterprojekt</b>
<b>Kürzel:</b>	MIN-MSPR
<b>Studiensemester:</b>	3. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Jürgen Dunkel
<b>Dozent(in):</b>	Alle Lehrenden
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, Pflichtfach, 3. Semester
<b>Lehrform/SWS:</b>	6 SWS Projektarbeit
<b>Arbeitsaufwand:</b>	Projektarbeit = 180 h Eigenstudium = 180 h
<b>Kreditpunkte:</b>	12 CP (= 360 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	
<b>Empfohlene Voraussetzungen:</b>	alle Lehrveranstaltungen der jeweiligen Vertiefungsrichtung, insbesondere Projekt- und Qualitäts-Management
<b>Angestrebte Lernergebnisse:</b>	Analyse-Kompetenzen: die gegebene Fragestellung unter Einsatz wissenschaftlicher Methoden und Erkenntnisse eigenständig analysieren können. Design- /Realisierungs-Kompetenzen: Lösungsstrategien entwickeln und diese mit den erworbenen Kenntnissen umzusetzen können. Technologische Kompetenzen: Wissen aus verschiedenen Bereichen kombinieren und zielgerichtet einsetzen können. Methoden-Kompetenzen: verschiedene innovative Methoden zur Lösung praktischer Probleme anwenden können. Projektmanagement-Kompetenz: Fähigkeit zur Projektplanung, zum Aufbau einer Organisationsstruktur und zur gezielten Steuerung von Projekten. Risiko-Management Soziale Kompetenzen: Anwendung von Konflikt-Lösungsstrategien
<b>Inhalt:</b>	Im Masterprojekt wird eine komplexe und aktuelle Fragestellung der Informatik bearbeitet, die einem der Schwerpunkte entstammen kann. Kennzeichnend ist dabei, dass die Fragestellung in einem Projektteam eigenständig mit projektspezifischer Organisation, Durchführung und Planung bearbeitet wird. Die konkreten Inhalte, Verfahren und Methoden ergeben sich aus der jeweiligen Fragestellung.
<b>Studien-/ Prüfungsleistungen:</b>	Experimentelle Arbeit
<b>Medienformen:</b>	Projektarbeit: Präsentation, Tafel, Beispielaufgaben, Diskussion Selbständige Bearbeitung der Aufgaben, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Projektspezifisch

<b>Modulbezeichnung:</b>	<b>Masterarbeit</b>
<b>Kürzel:</b>	MIN-MA
<b>Studiensemester:</b>	4. Semester (SoSe), Turnus: Jährlich (bei Bedarf auch jedes Semester)
<b>Modulverantwortliche(r):</b>	Stefan Wohlfeil
<b>Dozent(in):</b>	Alle Dozentinnen und Dozenten
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, Pflichtfach, 4. Semester
<b>Lehrform/SWS:</b>	
<b>Arbeitsaufwand:</b>	Eigenstudium = 810 h; Vorbereitung und Durchführung des Kolloquiums = 90 h
<b>Kreditpunkte:</b>	30 CP (= 900 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Zulassung zum Kolloquium nur wenn sämtliche Fachprüfungen bestanden
<b>Empfohlene Voraussetzungen:</b>	Alle Master-Module des 1.-3. Semesters gemäß gewählten Schwerpunkten.
<b>Angestrebte Lernergebnisse:</b>	Fähigkeit ein komplexes, praxisbezogenes Informatik-Thema selbstständig und mit wissenschaftlichen Methoden zu bearbeiten, d.h. Problemstellung analysieren, Lösungsmöglichkeiten aufzeigen, in den Stand der Wissenschaft/Technik einordnen und evtl. implementieren und abschließend bewerten. Fähigkeit zum Verfassen einer anspruchsvollen wissenschaftlichen Ausarbeitung zum Thema. Im Kolloquium zeigen die Studierenden, dass sie komplexe Sachverhalte in kurzer Zeit vor Fachpublikum verständlich präsentieren können.
<b>Inhalt:</b>	Im Gegensatz zur Bachelorarbeit wird hier ein anspruchsvolleres und evtl. umfangreicheres Thema auf einem wissenschaftlich höheren Niveau über einen längeren Zeitraum bearbeitet.
<b>Studien-/Prüfungsleistungen:</b>	Schriftliche Abschlussarbeit, eigener Vortrag und Fachdiskussion über das Thema der Arbeit / Verteidigung der Arbeit.
<b>Medienformen:</b>	
<b>Literatur:</b>	Wird abhängig vom Thema der Arbeit von den Dozenten vorgeschlagen und dann von den Studierenden selbstständig aktualisiert, erweitert und ergänzt.

## 2. Schwerpunkt Computergrafik

<b>Modulbezeichnung:</b>	<b>Geometrische Modellierung</b>
<b>Kürzel</b>	MIN-GM
<b>Studiensemester:</b>	2. Semester (SoSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Frauke Sprengel
<b>Dozent(in):</b>	Frauke Sprengel, Ingo Ginkel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 2. Semester, Modul im Schwerpunkt Computergrafik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Computergrafik 1, Mathematik 3 aus dem Bachelor-Studium
<b>Angestrebte Lernergebnisse:</b>	Theoretische und methodische Kompetenz: erwerben grundlegende Fachkenntnisse im Bereich Geometrische Modellierung, entwickeln Verständnis für anwendbare Techniken und Methoden und deren Grenzen Analyse-, Design- und Realisierungskompetenz: sind fähig, Modellierungsprobleme zu formulieren, zu formalisieren und zu lösen (d.h. entsprechende Algorithmen zu implementieren)
<b>Inhalt:</b>	Freiformkurven und -flächen (Bezier, B-Splines, NURBS), Unterteilungsflächen, Körpermodellierung (Constructive Solid Geometry), Geometrische Stetigkeit, Interpolation, Approximation
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispiele, Diskussion Übung: Selbständige Bearbeitung praktischer Aufgaben in Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript / Folien zur Vorlesung Gerald Farin: Curves and Surfaces in CAGD, neueste Auflage C. de Boor. A Practical Guide to Splines. <i>Springer</i> , Heidelberg 1987 H. Prautzsch: Bézier and B-Spline Techniques, Springer 2002

<b>Modulbezeichnung:</b>	<b>Computergrafik und Interaktion</b>
<b>Kürzel:</b>	MIN-CGI
<b>Studiensemester:</b>	1. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Volker Ahlers
<b>Dozent(in):</b>	Volker Ahlers, Frauke Sprengel, Ingo Ginkel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, 1. Semester, Modul im Schwerpunkt Computergrafik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Einführungsvorlesung zur Computergrafik im Bachelorstudium
<b>Angestrebte Lernergebnisse:</b>	<p>Algorithmische Kompetenzen: Verständnis der mathematischen und algorithmischen Grundlagen der Computergrafik, insb. des Echtzeit-Renderng, sowie der virtuellen und erweiterten Realität</p> <p>Analyse-, Design- und Realisierungskompetenzen: Design und Realisierung interaktiver Grafikanwendungen mit aktuellen Grafikbibliotheken; Design und Realisierung natürlicher Benutzerschnittstellen</p> <p>Technologische Kompetenzen: Verständnis der Funktionsweise moderner Grafikprozessoren; Kenntnis des Einsatzes der Computergrafik in unterschiedlichen Anwendungsgebieten</p> <p>Methodenkompetenzen: Kenntnis der Möglichkeiten, des Nutzens und der Grenzen des Einsatzes von Techniken der Computergrafik sowie der virtuellen und erweiterten Realität</p>
<b>Inhalt:</b>	<ul style="list-style-type: none"> <li>• Grundlagen: Echtzeit-Renderng, Architektur moderner Grafikprozessoren, Rendering Pipeline, programmierbare Shader, Konzepte der virtuellen und erweiterten Realität</li> <li>• Fortgeschrittene Rendering-Techniken: Spiegelung, Schatten, Image-based Rendering, Partikelsysteme, Kollisionserkennung</li> <li>• Modellierung: Szenengraphen, räumliche Datenstrukturen</li> <li>• Interaktion: natürliche Benutzerschnittstellen, Stereo Rendering, Motion Tracking, Ansteuerung entsprechender Peripheriegeräte</li> <li>• Software: Aktuelle Grafik-, Szenengraph- und VR-Bibliotheken, Anwendungen der Computergrafik</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Bearbeitung von Aufgaben in Gruppen, Projektarbeit mit Ergebnispräsentation, Begutachtung und Diskussion der Lösungen, Gesprächsführung.</p>
<b>Literatur:</b>	<p>Skript zur Vorlesung</p> <p>Akenine-Möller, T., Haines, E., Hoffman, N.: Real-Time Rendering. A K Peters, neueste Auflage</p> <p>Nischwitz, A., Fischer, M., Haberäcker, P., Socher, G.: Computergrafik und Bildverarbeitung, Band 1. Vieweg+Teubner, neueste Auflage</p>

<b>Modulbezeichnung:</b>	<b>Visual Computing</b>
<b>Kürzel</b>	MIN-VC
<b>Studiensemester:</b>	3. Semester (WiSe), Turnus: Jährlich, wechselnd mit MIN-COG und MIN-CV
<b>Modulverantwortliche(r):</b>	Frauke Sprengel
<b>Dozent(in):</b>	Volker Ahlers, Elisabeth Dennert-Möller, Frauke Sprengel, Ingo Ginkel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 3. Semester, Modul im Schwerpunkt Computergrafik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Grundkenntnisse in digitaler Bildverarbeitung und Computergrafik
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: tiefgehende Fachkenntnisse in einem ausgewählten Spezialgebiet des Visual Computing, Verständnis für anwendbare Methoden sowie deren Grenzen Methodenkompetenz: Anwendung innovativer Methoden im ausgewählten Bereich Analyse-, Design- und Realisierungskompetenz: Fähigkeiten, Probleme aus einem neuen und in der Entwicklung begriffenen Bereich der Computergrafik, Computer Vision und Visualisierung zu formulieren, zu formalisieren und zu lösen
<b>Inhalt:</b>	Es wird ein ausgewähltes Thema aus dem Bereich Visual Computing vorgestellt, z.B. Medizinische Visualisierung, digitale Bilderzeugung, Mustererkennung und maschinelles Lernen, Künstliche Intelligenz, Robotik, GPU-Computing, Bilddatenbanken, Auffinden von Ereignissen in Bildfolgen, Modellierung und Simulation. Dabei sollen vor allen Dingen aktuelle Themen und Entwicklungen Berücksichtigung finden. Forschungsnahe Fragestellungen legen dabei ggf. eine Basis für die Master-Arbeit.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, Projektarbeit, Gesprächsführung
<b>Literatur:</b>	Aktuell nach Lehrinhalten

<b>Modulbezeichnung:</b>	<b>Computational Geometry</b>
<b>Kürzel</b>	MIN-COG
<b>Studiensemester:</b>	3. Semester (WiSe), Turnus: Jährlich, wechselnd mit MIN-VC und MIN-CV
<b>Modulverantwortliche(r):</b>	Ingo Ginkel
<b>Dozent(in):</b>	Ingo Ginkel, Volker Ahlers, Frauke Sprengel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 3. Semester, Modul im Schwerpunkt Computergrafik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Geometrische Modellierung, Computergrafik und Interaktion
<b>Angestrebte Lernergebnisse:</b>	Theoretische und mathematische Kompetenzen: Charakterisierung des Schwierigkeitsgrads vorgegebener geometrischer Probleme sowie deren Klassifizierung bezüglich effizienter algorithmischer Lösbarkeit. Analyse-, Design- und Realisierungskompetenzen: Kenntnis verschiedener algorithmischer Lösungsstrategien zur Bearbeitung praxisrelevanter geometrischer Fragestellungen. Technologische Kompetenzen: Implementierung dieser Algorithmen
<b>Inhalt:</b>	Es werden ausgewählte Themen aus dem Bereich Computational Geometry vorgestellt, z.B. Art-Gallery-Problem, Polygon-Triangulation, Voronoi-Diagramme, Delaunay-Triangulation, Windowing, Point Location, Effiziente Kollisionsvermeidungsstrategien, Raumdatenstrukturen wie Octrees oder kd-Trees, Robot Motion-Planning, Graphentheorie und Graphen-Algorithmen, Point-based Graphics, Netzdatenverarbeitung, Geometrische Filteralgorithmen, etc. Dabei sollen vor allen Dingen aktuelle Themen und Entwicklungen Berücksichtigung finden. Forschungsnahe Fragestellungen legen dabei ggf. eine Basis für die Master-Arbeit.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, Projektarbeit, Gesprächsführung
<b>Literatur:</b>	De Berg: Computational Geometry, Springer 2008, sowie weitere Literatur aktuell nach Lehrinhalten

<b>Modulbezeichnung:</b>	<b>Computer Vision</b>
<b>Kürzel</b>	MIN-CV
<b>Studiensemester:</b>	3. Semester (WiSe), Turnus: Jährlich, wechselnd mit MIN-COG und MIN-VC
<b>Modulverantwortliche(r):</b>	Elisabeth Dennert-Möller
<b>Dozent(in):</b>	Elisabeth Dennert-Möller, Frauke Sprengel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 3. Semester, Modul im Schwerpunkt Computergrafik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung, Projektarbeit in Kleingruppen
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Projektarbeit und Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Grundlagen der digitalen Bildverarbeitung (siehe CG2 aus Bachelor – Studiengang)
<b>Angestrebte Lernergebnisse:</b>	Algorithmische und mathematische Kompetenz: profundes Wissen und Verständnis für die mathematischen und algorithmischen Prinzipien der digitalen Bildverarbeitung, Analyse – und Design - Kompetenz: Lösen von unüblich oder unvollständig definierten Problemen der Bildanalyse in verschiedenen Anwendungsszenarien Technologische Kompetenz durch Kombinieren von Wissen aus Bildverarbeitung, Mathematik und Informatik und Erkennen der Grenzen von Bildanalyseverfahren
<b>Inhalt:</b>	Vertiefte Kenntnisse über Verfahren zur Erkennung und Extraktion logisch zusammenhängender Bildinhalte in Bildern oder Bildfolgen und ihrer Beschreibung auf höherem Abstraktionsniveau, um bspw. abgebildete Objekte zu erkennen <ul style="list-style-type: none"> <li>- Segmentierungsverfahren</li> <li>- Repräsentation und Beschreibung von Segmenten</li> <li>- Objekterkennung</li> <li>- Charakterisierung und Vergleich von ähnlichen Bildern</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbstständige Bearbeitung von Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Skript zur Vorlesung Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Prentice Hall Jähne, B. Digitale Bildverarbeitung, Springer - Verlag Baggio, Escriva, Mahmood et al.: Mastering OpenCV with Practical Computer Vision Projects, [packt]publishing



## 4. Schwerpunkt IT-Sicherheit

<b>Modulbezeichnung:</b>	<b>IT-Sicherheit I</b>
<b>Kürzel:</b>	MIN-ITS1
<b>Studiensemester:</b>	1. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Stefan Wohlfeil
<b>Dozent(in):</b>	Stefan Wohlfeil, Josef von Helden, Holger Peine, Matthias Hovestadt
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, 1. Semester, Modul im Schwerpunkt IT-Sicherheit
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Kenntnisse von Betriebssystem- und Netz-Technologien
<b>Angestrebte Lernergebnisse:</b>	<p>Technologische Kompetenzen: Die Studierenden besitzen fundierte, tiefgehende Kenntnisse von Sicherheitsmaßnahmen u. -Mechanismen. Sie können Wissen aus den Bereichen Mathematik, Rechnernetze und IT-Sicherheit kombinieren und mit der deren Komplexität umgehen. Sie kennen die Grundlagen der wichtigsten Techniken im Bereich IT-Sicherheit und können diese selbständig erweitern und vertiefen.</p> <p>Design- und Realisierungskompetenzen: Die Studierenden können unübliche, unvollständig definierte Probleme aus dem Bereich IT-Sicherheit lösen und die Lösungen auch technisch umsetzen.</p> <p>Soziale Kompetenzen: Die Studierenden erwerben in den Übungen die Fähigkeit zu eigenverantwortlicher Tätigkeit (Einarbeitung, Analyse, Entwurf und Umsetzung) sowie zur Zusammenarbeit in Teams.</p>
<b>Inhalt:</b>	<p>Fachterminologie: Bedrohungen, Risiken, Schwachstellen, Schutzziele.</p> <p>Sicherheitstechniken: Authentisierung, Autorisierung, Identity Management; Kryptographie; Hash-Funktionen; Message Authentication Codes; Digitale Signaturen; X.509 Zertifikate; Anwendung der Techniken bei E-Mail (PGP, S/MIME), WWW (SSL), Anmeldung auf entfernten Rechnern (SSH); Kerberos; Sicherheitsstandards/-prozesse; Forensik; Web-Sicherheit.</p>
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispiele, Diskussion</p> <p>Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung</p>
<b>Literatur:</b>	<p>C. Eckert: IT- Sicherheit. Konzepte, Verfahren, Protokolle; Oldenbourg</p> <p>C. Kaufman, R. Perlman, M. Speciner: Network Security – Private Communication in a PUBLIC World; Prentice Hall</p> <p>W. Stallings: Cryptography and Network Security; Prentice-Hall</p> <p>S. Wohlfeil: "Kurs 1866 - Sicherheit im Internet", FernUniversität Hagen</p>

<b>Modulbezeichnung:</b>	<b>IT-Sicherheit II</b>
<b>Kürzel</b>	MIN-ITS2
<b>Studiensemester:</b>	2. Semester (SoSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Stefan Wohlfeil
<b>Dozent(in):</b>	Stefan Wohlfeil, Josef von Helden, Holger Peine, Matthias Hovestadt
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 2. Semester, Modul im Schwerpunkt IT-Sicherheit
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	IT-Sicherheit 1
<b>Angestrebte Lernergebnisse:</b>	<p>Technologische Kompetenzen: Die Studierenden besitzen fundierte und tiefgehende Kenntnisse von Sicherheitsmaßnahmen und -Mechanismen. Sie können Wissen aus den Bereichen Mathematik, Rechnernetze und IT-Sicherheit kombinieren und mit der entstehenden Komplexität umgehen. Sie kennen die Grundlagen der wichtigsten Techniken im Bereich IT-Sicherheit und können diese selbständig erweitern und vertiefen.</p> <p>Design- und Realisierungskompetenzen: Die Studierenden können unübliche, unvollständig definierte Probleme aus dem Bereich IT-Sicherheit lösen und die Lösungen auch technisch umsetzen.</p> <p>Soziale Kompetenzen: Die Studierenden erwerben in den Übungen die Fähigkeit zu eigenverantwortlicher Tätigkeit (Einarbeitung, Analyse, Entwurf und Umsetzung) sowie zur Zusammenarbeit in Teams.</p>
<b>Inhalt:</b>	Sicherheit im E-Commerce, sichere Bezahlverfahren; Biometrie; Anonymität und Privacy (Mixe); Intrusion Detection (Prinzipien und netzbasierte sowie hostbasierte Systeme); Virtual Private Networks (Funktionsweise, Eigenschaften, Systeme wie IPSEC, OpenVPN), Sicherheitsarchitekturen, Firewallarchitekturen (Paketfilter, ALG, DMZ, etc.); Virtualisierung;
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispiele, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	S. Northcutt, J. Novak: Network Intrusion Detection; New Riders J. Snader: VPNs Illustrated; Addison-Wesley S. Wohlfeil: "Kurs 1867 - Sicherheit im Internet 2"; FernUniversität Hagen.

<b>Modulbezeichnung:</b>	<b>Spezialthema IT-Sicherheit (IT-Sicherheit III)</b>
<b>Kürzel:</b>	MIN-SSI
<b>Studiensemester:</b>	3. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Josef von Helden
<b>Dozent(in):</b>	Josef von Helden, Stefan Wohlfeil, Holger Peine
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, 3. Semester, Modul im Schwerpunkt IT-Sicherheit
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	IT-Sicherheit I und II
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: tiefgehende Fachkenntnisse in einem ausgewählten Spezialgebiet der IT-Sicherheit erwerben; Verständnis für anwendbare Techniken und Methoden und deren Grenzen entwickeln. Methodenkompetenz: innovative Methoden des ausgewählten Bereichs anwenden können; Voraussetzungen für die Weiterentwicklung der IT-Sicherheit als wissenschaftliche Disziplin erwerben Übergreifend: Transferkompetenz; komplexe Zusammenhänge erfassen und zusammenfassen können
<b>Inhalt:</b>	Ausgewählte aktuelle Themen aus dem Bereich der IT-Sicherheit, z.B. Trusted Computing, Kryptographie, Entwicklung sicherer Software, Sicherheitsbewertung, -überprüfung und -management, Bedrohungs- und Risikoanalysen, Penetrationstests.
<b>Studien-/Prüfungsleistungen:</b>	Mündliche Prüfung und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Demonstration, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, teilweise in Zusammenarbeit mit anderen Gruppen, Begutachtung der Lösungen, Zusammenfassung der Ergebnisse, Gesprächsführung
<b>Literatur:</b>	Aktuell nach Lehrinhalten.

<b>Modulbezeichnung:</b>	<b>Secure Software Engineering</b>
<b>Kürzel</b>	MIN-SSE
<b>Studiensemester:</b>	1., 2. oder 3., in der Regel Jährlich
<b>Modulverantwortliche(r):</b>	Holger Peine
<b>Dozent(in):</b>	Holger Peine, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 1.-3. Semester, Modul im Schwerpunkt IT-Sicherheit
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	IT-Sicherheit 1, gute Kenntnisse der typischen Aktivitäten eines Softwareentwicklungsprozesses (einschließlich Programmierung in Java und C), Kenntnisse der Funktion verteilter Anwendungen
<b>Angestrebte Lernergebnisse:</b>	<ul style="list-style-type: none"> <li>• Analysekompetenzen und Technologische Kompetenzen: Studierende können die wichtigsten Klassen von Software-Sicherheitsschwachstellen in realen Systemen erkennen und ihr Schadenspotential einschätzen</li> <li>• Design-, Realisierungs- und Methodenkompetenzen: Studierende beherrschen Techniken und Methoden für verschiedene Aktivitäten im Softwareentwicklungsprozess, um dort Schwachstellen zu vermeiden oder aufzudecken</li> </ul>
<b>Inhalt:</b>	<ul style="list-style-type: none"> <li>• Klassen von Sicherheitsschwachstellen in Software und ihre Ursachen</li> <li>• Erhebung und Formulierung von Sicherheitsanforderungen</li> <li>• Bedrohungen Finden, Einschätzen und Abwehren</li> <li>• Software-Sicherheit auf Architektur-Ebene</li> <li>• Software-Sicherheit auf Quelltext-Ebene</li> <li>• Testen und Inspektion auf Sicherheitsschwachstellen</li> <li>• Benutzerfreundlichkeit und Prozessmodelle bzgl. Sicherheit</li> </ul>
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, experimentelle Projektarbeit, Gesprächsführung
<b>Literatur:</b>	Michael Howard, David LeBlanc: Writing Secure Code, 2 <sup>nd</sup> ed. 2003 Gary McGraw: Software Security - Building Security In, 2006 John Viega, Gary McGraw: Building Secure Software, 2001 Michael Howard, Steve Lipner: The Security Development Lifecycle, 2006 Sachar Paulus: Basiswissen Sichere Software, 2011.

## 5. Schwerpunkt Informationssysteme

<b>Modulbezeichnung:</b>	<b>Datenbankparadigmen</b>
<b>Kürzel</b>	MIN-DBP
<b>Studiensemester:</b>	1. oder 3. Semester (WiSe), Turnus: Jährlich
<b>Modulverantwortliche(r):</b>	Felix Heine
<b>Dozent(in):</b>	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner, Arne Koschel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Informationssysteme
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Keine
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse- und Design-Kompetenz: Verwendungsmöglichkeiten alternativer DB-Paradigmen erkennen; das richtige Modell bzw. das richtige System für das gegebene Problem erkennen; Anwendungen auf dieser Basis konzeptionieren können.</p> <p>Algorithmische Kompetenzen: moderne DB-Algorithmen, insbesondere Speicherorganisation und Anfrageverarbeitung, kennen und beurteilen können. Vor- und Nachteile der Algorithmen für gegebene Problemstellungen erkennen.</p> <p>Technologische Kompetenzen: Abfragesprachen beherrschen, Zugriffsschnittstellen beherrschen.</p>
<b>Inhalt:</b>	<p>Schwerpunktmäßig werden in dem Modul Alternativen zu relationalen Datenbank behandelt, sowie fortgeschrittene Erweiterungen relationaler Datenbanken. Dabei wird sowohl auf die Datenmodellierung und die Abfragesprachen als auch auf Implementierungsaspekte wie z.B. Anfragebearbeitung eingegangen. Ggf. wird auch die Anwendungsprogrammierung auf Basis dieser DB-Paradigmen betrachtet. Themen können dabei z.B. sein:</p> <ul style="list-style-type: none"> <li>• Objektrelationale- und Objektorientierte Datenbanken</li> <li>• XML-Datenbanken und Abfragesprachen</li> <li>• Graphdatenbanken</li> <li>• Key/Value-Datenbanken</li> <li>• Dokumenten-Orientierte Datenbanken</li> <li>• Spaltenorientierte Speicherung von Daten</li> <li>• In-Memory-Datenbanken</li> <li>• Embedded Datenbanken</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation &amp; Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit</p>
<b>Literatur:</b>	<p>Edlich, Friedland, Hampe, Brauer: „N*SQL. Einstieg in die Welt Nichtrelationaler Web 2.0 Datenbanken“, Hanser</p> <p>Dietrich, Urban: „An Advanced Course in Database Systems: Beyond Relational Databases“, Prentice Hall</p> <p>Weitere aktuelle Quellen zu den Vorlesungsinhalten.</p>

<b>Modulbezeichnung:</b>	<b>Verteilte Informationssysteme</b>
<b>Kürzel</b>	MIN-VIS
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Arne Koschel
<b>Dozent(in):</b>	Arne Koschel, Felix Heine, Matthias Hovestadt
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Informationssysteme
<b>Lehrform/SWS:</b>	4 SWS: Vorlesung, Projekt und Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Abgeschlossenes Bachelor-Studium mit Informatikanteil, der mindestens den Programmieren-, Informationssysteme- und Betriebssysteme/Netze-Veranstaltungen vergleichbar ist.
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse und Design-Kompetenzen: Die Studierenden sollen Informationssysteme hinsichtlich behandelter Kriterien beurteilen bzw. entwerfen und umsetzen können. Muster auswählen und anwenden können.</p> <p>Algorithmische Kompetenzen: Verteilte Algorithmen verstehen und einsetzen können.</p> <p>Technologische Kompetenzen: Die Studierenden sollen Kenntnisse über Konzepte und Technologien zu Aufbau, Betrieb und Nutzung verteilter Informationssysteme erlangen. Im Kontext verschiedener Architekturmodelle sollen hierzu für verteilte Informationssysteme notwendige Aspekte betrachtet werden.</p>
<b>Inhalt:</b>	<p>In diesem Modul werden fortgeschrittene Kompetenzen des Aufbaus und Betriebs verteilter Informationssysteme vermittelt. Hierzu werden Kernanforderungen an solche Informationssysteme beleuchtet und Verfahren zur Erfüllung dieser Anforderungen vorgestellt. Theoretische Konzepte werden hierbei im Kontext existierender Softwaresysteme erlernt und angewandt. Wichtige Aspekte hierbei betreffen:</p> <ul style="list-style-type: none"> <li>• Grundlegende Architekturkonzepte zur Realisierung verteilter Informationssysteme (Cluster, Cloud, ...);</li> <li>• Datenintegration (fDBMS, Datenintegrationstools, ...);</li> <li>• Aspekte der Parallelität (Skalierbarkeit, Lastbalancierung, Konsistenz, weiterführenden Transaktionskonzepte, ...);</li> <li>• und weitere aktuelle Themen wie z.B. BigData</li> </ul>
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation &amp; Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit</p>
<b>Literatur:</b>	<p>Conrad, S., Hasselbring, W., Koschel, A., Tritsch, R.: „Enterprise Application Integration“, Spektrum/Elsevier.</p> <p>Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., Koschel, A. „Systemarchitekturen für verteilte Anwendungen“, Hanser.</p> <p>Tanenbaum, A.: „Verteilte Systeme: Prinzipien und Paradigmen“, Pearson.</p> <p>Weitere aktuelle Quellen zu den Vorlesungsinhalten.</p>

<b>Modulbezeichnung:</b>	<b>Datenanalyse</b>
<b>Kürzel</b>	MIN-DA
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Felix Heine
<b>Dozent(in):</b>	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner, Arne Koschel
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Informationssysteme
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Keine
<b>Angestrebte Lernergebnisse:</b>	Analyse- und Design-Kompetenz: Verfahren zum Auffinden und Analysieren von Daten kennen und einsetzen können. Anhand des gegebenen Problems das richtige Verfahren oder die richtige Kombination aus Verfahren auswählen können. Architektur eines Data Warehouse verstehen. Algorithmische Kompetenzen: Analyseverfahren verstehen, Vorbedingungen, Einsatzgebiete sowie Vor- und Nachteile kennen. Technologische Kompetenzen: Beispiel-Systeme wie z.B. Rapid-Miner nutzen können.
<b>Inhalt:</b>	Es werden verschiedene Herangehensweisen zur Organisation und Analyse großer Datenbestände beschrieben. Dabei stehen folgende Bereiche im Vordergrund: <ul style="list-style-type: none"> <li>• Data Warehousing</li> <li>• Data Mining</li> <li>• Datenqualität</li> <li>• Information Retrieval</li> </ul> Anhand von exemplarischen Systemen wie z.B. MS SQL Server, Rapid Miner, etc. werden die Verfahren in den Übungen praktisch eingesetzt.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation & Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit
<b>Literatur:</b>	Bauer, Günzel: "Data Warehouse Systeme", dpunkt.verlag Tan, Steinbach, Kumar: „Introduction to Data Mining“, Addison Wesley Manning, Prabhakar Raghavan, Schütze: "Introduction to Information Retrieval", Cambridge University Press Schmitt: "Ähnlichkeitssuche in Multimedia-Datenbanken", Oldenbourg Weitere aktuelle Quellen zu den Vorlesungsinhalten.

<b>Modulbezeichnung:</b>	<b>Geo- und Multimedia-Informationssysteme</b>
<b>Kürzel:</b>	MIN-GMI
<b>Studiensemester:</b>	2.(SoSe) oder 3. Semester (WiSe), Turnus: in der Regel alle zwei Jahre
<b>Modulverantwortliche(r):</b>	Carsten Kleiner
<b>Dozent(in):</b>	Elisabeth Dennert-Möller, Felix Heine, Carsten Kleiner
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, 2.-3. Semester, Modul im Schwerpunkt Informationssysteme
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Kenntnisse aus einem Bachelorstudium im Umfang der Module IS1, IS2
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse-, Design- und Realisierungskompetenzen: Stärken und Schwächen verschiedener Möglichkeiten zur persistenten Speicherung zwei- und drei-dimensionaler Daten kennenlernen und bewerten können; Modellierung und Anfragetypen für Multimediadaten kennenlernen, bewerten und vergleichen können; für genannte Daten beste persistente Speichervariante in bestimmtem Anwendungsfeld auswählen und realisieren können</p> <p>Algorithmische Kompetenzen: Verschiedene geometrische Algorithmen verstehen, deren Laufzeiten und Einsatzmöglichkeiten bewerten können sowie diese effizient implementieren können.</p> <p>Technologische Kompetenzen: Speicherung und Abfrage räumlicher Daten unterschiedlicher Modelle in Datenbanksystemen beherrschen und einsetzen können (Oracle Spatial, PostGIS); Speicherung und Abfrage von Multimediadaten in Datenbanksystemen beherrschen und einsetzen können (Oracle, Postgres); alternative Datenbankparadigmen für diese Datentypen kennen.</p>
<b>Inhalt:</b>	Räumliche Datentypen und geometrische Algorithmen, Datenbank- und Informationssysteme für räumliche Daten (z.B. Geoinformationssysteme), drei-dimensionale Datenmodelle und deren Speicherung, Persistente Speicherung und Abfrage von Multimediadaten Zugriffsstrukturen für spezielle Anwendungsbereiche.
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Beispielaufgaben, Diskussion, Visualisierung Übung: Bearbeitung von Übungsaufgaben, Besprechung der Ergebnisse (Tafel, Gespräch), Bearbeitung einer Projektaufgabe mit Präsentation der Ergebnisse, Erstellung eines schriftlichen Berichts
<b>Literatur:</b>	De Berg, van Kreveld, Overmars: Computational Geometry, Springer Rigaux, Scholl, Voissard: Spatial Databases with application to GIS, Morgan Kaufmann T. Brinkhoff: Geodatenbanksysteme in Theorie und Praxis, Wichmann I. Schmitt: Ähnlichkeitssuche in Multimedia-Datenbanken, Oldenbg.



## 6. Schwerpunkt Softwaretechnik

<b>Modulbezeichnung:</b>	<b>Programmierparadigmen</b>
<b>Kürzel</b>	MIN-PPD
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Holger Peine
<b>Dozent(in):</b>	Holger Peine, Robert Garmann, Carsten Kleiner, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch oder Englisch
<b>Einordnung Curriculum</b>	Master, Wahlpflichtfach, 1.-3. Semester, Schwerpunkt Softwaretechnik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen:</b>	Keine
<b>Empfohlen:</b>	Gute Kenntnisse in objektorientierter Programmierung
<b>Angestrebte Lernergebnisse:</b>	<ul style="list-style-type: none"> <li>• Analyse-, Design- und Realisierungs-Kompetenzen: Einordnen und Einschätzen der Ausdrucksmittel von Programmiersprachen</li> <li>• Methodenkompetenzen: Erweiterung der Problemlösefähigkeiten durch Beherrschung alternativer Denkansätze, Vertiefung des Verständnisses der objektorientierten Programmierung</li> <li>• Technologische Kompetenzen: Erlernen einer funktionalen Programmiersprache (z. B. Scala) und (in begrenztem Umfang) einer logischen Programmiersprache (z. B. Prolog)</li> </ul>
<b>Inhalt:</b>	<ul style="list-style-type: none"> <li>• Programmiersprachen und -paradigmen (kurz)</li> <li>• Imperatives Paradigma (kurz)</li> <li>• Objektorientiertes Paradigma (mittellang) <ul style="list-style-type: none"> <li>○ Kapselung, Vererbung</li> <li>○ Polymorphismus, Substitutionsprinzip, Varianz</li> </ul> </li> <li>• Funktionales Paradigma (lang) <ul style="list-style-type: none"> <li>○ Unveränderlichkeit und Seiteneffektfreiheit</li> <li>○ Funktionsobjekte, Funktionen höherer Ordnung, Closures</li> <li>○ Verzögerte Auswertung, Continuations, funktionaler Zustand</li> </ul> </li> <li>• Logisches Paradigma (mittellang) <ul style="list-style-type: none"> <li>○ Aussagenlogik, Prädikatenlogik, Hornlogik, Resolution</li> <li>○ Unifikation, negation as failure, prozedurale Semantik</li> <li>○ Optional: Typisierung, Integration mit anderen Paradigmen</li> </ul> </li> </ul>
<b>Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Experimentelle Arbeit (einzeln und in Gruppen), Diskussion
<b>Literatur:</b>	Michael L. Scott: Programming Language Pragmatics Bertrand Meyer: Object oriented software construction P. Chiusano, R. Bjarnason: Functional Programming in Scala Ivan Bratko: PROLOG Programming for Artificial Intelligence.

<b>Modulbezeichnung:</b>	<b>Intelligente Systeme</b>
<b>Kürzel:</b>	MIN-IS
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Ralf Bruns
<b>Dozent(in):</b>	Ralf Bruns, Jürgen Dunkel, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Softwarearchitektur
<b>Angestrebte Lernergebnisse:</b>	Technologische Kompetenz: Erwerb tiefgehender Fachkenntnisse im Gebiet Intelligente Systeme (Künstliche Intelligenz), Verständnis entwickeln für anwendbare Techniken und Methoden; Nutzen und Grenzen intelligenter Technologien einschätzen können Methodenkompetenz: Anwendung und kritische Beurteilung innovativer Methoden im Bereich Intelligenter Systeme Analyse-, Design- und Realisierungskompetenz: Fähigkeit, Probleme aus einem neuen Anwendungsbereich zu formulieren, zu strukturieren und mit Methoden intelligenter Systeme zu lösen
<b>Inhalt:</b>	Theoretische Grundlagen und Konzepte für Entwurf und Entwicklung von intelligenten (Anwendungs-)Systemen, z.B. Einführung in Künstliche Intelligenz, Multi-Agentensysteme, Inferenzalgorithmen, Semantic Web, Complex Event Processing Dabei sollen i.W. aktuelle Themen und Entwicklungen Berücksichtigung finden, insbesondere aus dem Bereich webbasierter und mobiler Systeme. Forschungsnahe Fragestellungen legen dabei ggf. eine Basis für die Masterarbeit.
<b>Studien-/ Prüfungsleistungen:</b>	Mündliche Prüfung und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in 2er-Gruppen, Begutachtung der Lösungen, Gesprächsführung
<b>Literatur:</b>	Aktuell nach Lehrinhalten

<b>Modulbezeichnung:</b>	<b>Fortgeschrittene Aspekte verteilter Systeme</b>
<b>Kürzel</b>	MIN-FAVS
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Arne Koschel
<b>Dozent(in):</b>	Arne Koschel, Felix Heine, Matthias Hovestadt
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
<b>Lehrform/SWS:</b>	4 SWS: Vorlesung, Projekt und Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Abgeschlossenes Bachelor-Studium mit Informatikanteil, der mindestens den Programmieren-, Informationssysteme- und Betriebssysteme/Netze-Veranstaltungen vergleichbar ist.
<b>Angestrebte Lernergebnisse:</b>	<p>Analyse und Design-Kompetenzen: Herausforderungen verschiedener Formen verteilter Systeme einschätzen und ihnen mithilfe geeigneter Verfahren und Technologien begegnen. Muster auswählen und anwenden können.</p> <p>Algorithmische Kompetenzen: Verteilte Algorithmen verstehen und einsetzen können.</p> <p>Technologische Kompetenzen: Die Studierenden sollen vertiefte Kenntnisse über Herausforderungen, Verfahren und aktuelle Forschungsergebnisse verschiedener Ansätze verteilter Systeme erlangen.</p>
<b>Inhalt:</b>	In diesem Modul werden fortgeschrittene Kompetenzen zu Architektur und Funktionalität verschiedener Formen verteilter Systemen vermittelt. Hierzu werden zunächst grundlegende Modelle für verteilte Systeme - z.B. Cloud Computing, P2P, SOA - beleuchtet und auf charakteristische Eigenschaften in Architektur, Verwaltung und Laufzeitverhalten sowie auf die Themenbereiche Parallelität und Fehlertoleranz eingegangen. Ausgewählte theoretische Konzepte werden mithilfe verschiedener Softwaresysteme praktisch umgesetzt. Hinzu kommen aktuelle Forschungsergebnisse und Fragestellungen.
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	<p>Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion</p> <p>Übung: Selbständige Aufgabenbearbeitung in kleinen Gruppen, Präsentation &amp; Begutachtung der Lösungen, Gesprächsführung, experimentelle Projektarbeit</p>
<b>Literatur:</b>	<p>Conrad, S., Hasselbring, W., Koschel, A., Tritsch, R.: „Enterprise Application Integration“, Spektrum/Elsevier.</p> <p>Dunkel, J., Eberhart, A., Fischer, S., Kleiner, C., Koschel, A. „Systemarchitekturen für verteilte Anwendungen“, Hanser.</p> <p>Tanenbaum, A.: „Verteilte Systeme: Prinzipien und Paradigmen“, Pearson.</p> <p>Weitere aktuelle Quellen zu den Vorlesungsinhalten.</p>

<b>Modulbezeichnung:</b>	<b>Software-Test und Requirements</b>
<b>Kürzel:</b>	MIN-STR
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Robert Garmann
<b>Dozent(in):</b>	Robert Garmann, Arne Koschel, Holger Peine
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum:</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
<b>Lehrform/SWS:</b>	4 SWS: Vorlesung, Projekt und Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h Übung = 34 h Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung:</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	Praktische Projekterfahrungen in allen Phasen des Softwareentwicklungsprozesses, insb. bei der Erstellung eines Pflichtenheftes und der Erstellung von verschiedenen Testroutinen Kenntnis verschiedener Testverfahren (z. B. black box / white box) in den verschiedenen Teststufen (Komponenten- bis Abnahmetest) Kenntnis der Grundbegriffe des Projekt- und Qualitätsmanagements
<b>Angestrebte Lernergebnisse:</b>	Analyse- und Design-Kompetenzen: Die Studierenden kennen grundlegende Prinzipien und Verfahren der Anforderungsanalyse und können diese in einem konkreten Anforderungsumfeld anwenden. Sie besitzen einen detaillierten Einblick in alle wesentlichen Testverfahren und können für ein komplexes System Testfälle entwerfen und die erwartete Qualität nachweisen. Sie können ein System hinsichtlich seiner fachlichen und technischen Qualitätsmerkmale bewerten. Realisierungs-Kompetenz: Die Studierenden können technische Testaktivitäten durchführen sowie die notwendigen Werkzeuge und Techniken zum Erreichen der definierten Testziele implementieren.
<b>Inhalt:</b>	Requirements: System und Systemkontext, Anforderungen ermitteln, (natürlichsprachlich) dokumentieren und konsolidieren, Anforderungen prüfen und abstimmen Software-Test: Testprozess, Testverfahren (spezifikationsorientiert, strukturorientiert, fehlerbasiert, erfahrungsbasiert), Test der Softwareeigenschaften bei fachlichen und technischen Tests, Testwerkzeuge und Testautomatisierung
<b>Studien-/ Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Diskussion, Aufsatzbesprechung Übung: Selbständige Bearbeitung von Aufgaben, experimentelle Projektarbeit, Referate, Gesprächsführung
<b>Literatur:</b>	Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering, dpunkt (2010) Andreas Spillner, Tilo Linz: Basiswissen Softwaretest, dpunkt (2005) Graham Bath, Judy McKay: Praxiswissen Softwaretest, dpunkt (2011)

<b>Modulbezeichnung:</b>	<b>Secure Software Engineering</b>
<b>Kürzel</b>	MIN-SSE
<b>Studiensemester:</b>	WiSe oder SoSe, Turnus in der Regel alle 1-2 Jahre
<b>Modulverantwortliche(r):</b>	Holger Peine
<b>Dozent(in):</b>	Holger Peine, Stefan Wohlfeil
<b>Sprache:</b>	Deutsch oder Englisch
<b>Zuordnung zum Curriculum</b>	Master, 1.-3. Semester, Modul im Schwerpunkt Softwaretechnik
<b>Lehrform/SWS:</b>	2 SWS Vorlesung, 2 SWS Übung
<b>Arbeitsaufwand:</b>	Vorlesung = 34 h, Übung = 34 h, Eigenstudium = 112 h
<b>Kreditpunkte:</b>	6 CP (= 180 h)
<b>Voraussetzungen nach Prüfungsordnung</b>	Keine
<b>Empfohlene Voraussetzungen:</b>	IT-Sicherheit 1, gute Kenntnisse der typischen Aktivitäten eines Softwareentwicklungsprozesses (einschließlich Programmierung in Java und C), Kenntnisse der Funktion verteilter Anwendungen
<b>Angestrebte Lernergebnisse:</b>	<ul style="list-style-type: none"> <li>• Analysekompetenzen und Technologische Kompetenzen: Studierende können die wichtigsten Klassen von Software-Sicherheitsschwachstellen in realen Systemen erkennen und ihr Schadenspotential einschätzen</li> <li>• Design-, Realisierungs- und Methodenkompetenzen: Studierende beherrschen Techniken und Methoden für verschiedene Aktivitäten im Software-Entwicklungsprozess, um dort Schwachstellen zu vermeiden oder aufzudecken</li> </ul>
<b>Inhalt:</b>	<ul style="list-style-type: none"> <li>• Klassen von Sicherheitsschwachstellen in Software und ihre Ursachen</li> <li>• Erhebung und Formulierung von Sicherheitsanforderungen</li> <li>• Bedrohungen Finden, Einschätzen und Abwehren</li> <li>• Software-Sicherheit auf Architektur-Ebene</li> <li>• Software-Sicherheit auf Quelltext-Ebene</li> <li>• Testen und Inspektion auf Sicherheitsschwachstellen</li> <li>• Benutzerfreundlichkeit und Prozessmodelle bzgl. Sicherheit</li> </ul>
<b>Studien-/Prüfungsleistungen:</b>	Prüfung (Klausur oder mündliche Prüfung) und experimentelle Arbeit
<b>Medienformen:</b>	Vorlesung: Präsentation, Tafel, Beispielaufgaben, Diskussion Übung: Selbständige Bearbeitung der Aufgaben in Gruppen, experimentelle Projektarbeit, Gesprächsführung
<b>Literatur:</b>	Michael Howard, David LeBlanc: Writing Secure Code, 2 <sup>nd</sup> ed. 2003 Gary McGraw: Software Security - Building Security In, 2006 John Viega, Gary McGraw: Building Secure Software, 2001 Michael Howard, Steve Lipner: The Security Development Lifecycle, 2006 Sachar Paulus: Basiswissen Sichere Software, 2011.